[redacted]

Report No. 473 Math COO-2118-0021

ILLIAC III REFERENCE MANUAL

VOLUME III: Input/Output

edited by
B. H. McCormick and B. J. Nordmann, Jr.

D. E. Atkins, R. T. Borovec, L. N. Goyal, L. M. Katoh
R. M. Lansford, J. C. Schwebel and V. G. Tareski

August 13, 1971

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

COO-2118-0021

Report No. 473

ILLIAC III REFERENCE MANUAL

VOLUME III: Input/Output

edited by

B. H. McCormick and B. J. Nordmann, Jr.

D. E. Atkins, R. T. Borovec, L. N. Goyal, L. M. Katoh,
R. M. Lansford, J. C. Schwebel and V. G. Tareski

August 13, 1971

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

This work was supported by Contract AT(11-1)-1018 with the U. S. Atomic Energy Commission through September 30, 1970. Current support is under Contract AT(11-1)-2118 with the above agency.

ABSTRACT

The Illiac III Reference Manual is being issued in this final documentation as four volumes:

	Volume I:	The Computer System
	Volume II:	Instruction Repertoire
This issue--	Volume III:	Input/Output
	Volume IV:	Supervisor Organization

For ease of cross-reference an integrated table of contents will be issued separately.

This third volume describes the input/output command repertoire of the Illinois Pattern Recognition Computer (ILLIAC III). Input/Output commands are used to control the transmission of data between the main store and peripheral devices. I/O devices considered here include conventional secondary storage units such as disks, tapes, etc., and image-oriented communication terminals such as scanners, monitors and elements of the video communication net.

Considerable autocracy of operation has been granted to the eight (maximum) channels associated with each of two possible Input/Output Processors of the Illiac III Computer system. Major design constraints have been (i) a provision for high data-rate streaming of scanner/camera-generated image data, and (ii) retention interface signaling conventions compatible with the IBM 360 series of computers.

ACKNOWLEDGMENTS

The Illiac III Reference Manual is based in part upon an earlier report:

(1) B.H. McCormick (editor), William D. Bond, Kimio Ibuki, Roger E. Wiegel and John A. Wilber, Preliminary Programming Manual for the Illiac III Computer, Department of Computer Science Report 185, University of Illinois, July 1965.

(2) B.H. McCormick and R.M. Lansford (editors), D.E. Atkins, R.T. Borovec, G.N. Cederquist, S.K.Chan, L.A. Dunn, J.P. Hayes, L.M. Katoh, P.L. Koo, B.J. Nordmann, Jr., J.A. Rohr, and J.C. Schwebel, Illiac III Programming Manual, Department of Computer Science Manual, University of Illinois, March 1968.

The present editors (B.H. McCormick and B.J. Nordmann, Jr.) acknowledge with gratitude the contribution of these earlier groups as transplanted to this greatly expanded manual. In addition, Dr. Rangaswamy Narasimhan contributed significantly to the early definition of the pattern articulation unit instruction set. In like manner Philip Merryman assisted materially in the early formulation of machine-implemented macros--the imprimitive instructions of Illiac III.

Authors and contributors to the manual are listed by principal area of concern:

Taxicrinic Processor:	B.J. Nordmann, Jr., R.M. Lansford, J.C. Schwebel, R.E. Wiegel
Input/Output Processor:	L.M. Katoh, V.G. Tareski, J.V. Wenta, G.N. Cederquist, J.P. Hayes
Arithmetic Units:	D.E. Atkins, L.N. Goyal
Pattern Articulation Unit:	R.T. Borovec, R.P. Harms, G.T. Lewis
Interrupt Unit:	L.M. Goyal
Exchange Net:	S.K. Chan, P. Krabbe
Scanner-Monitor-Video	L.A. Dunn, L.N. Goyal, V.G. Tareski, R.G. Martin, R.C. Amendola
Supervisor Organization:	B.J. Nordmann, Jr., R. M. Lansford

The seemingly endless drafts and revisions of this manual have been handled with patience and fortitude by Mrs. Betty Gunsalus and Mrs. Roberta Andre'. Illustrations were prepared by Mr. Stanley Zundo.

3. INPUT/OUTPUT COMMANDS

Input/output commands are used to control the transmission of data between the main store and peripheral devices. I/O devices considered include secondary storage units such as disks, tapes, etc., and communication terminals such as scanners, monitors, and elements of the video communication net.

In this manual, an instruction is executed in a TP, a command is executed in an IØP, and an order is executed by a peripheral device or device controller.

I/O Commands are subdivided into three main classes:

1) Interrupt Commands (IC's)

Interrupt commands are used to initiate, test and occasionally terminate execution of an I/O program. Upon instigation by supervisory program being executed in a taxicrinic processor, an interrupt command is transferred as an operand from the supervising TP to the IOP. The interrupt command is then interpreted and executed by the IOP exclusively.

2) Device Dependent Commands (DDC's)

A device dependent command specifies some operation to be carried out by a peripheral device/device controller connected to one of the two I/O Processors.

Device dependent commands are interpreted partly by the IOP and partly by the device to which the IOP transmits a portion of the command. Such commands are used to initiate data transfer operations with a particular device.

3) Device Independent Commands (DIC's)

Device independent commands are executed by the IOP only and usually concern bookkeeping operations e.g., loading or storing the contents of a program register. A DIC does not make reference to any particular device.

3.1 Interrupt Commands and Associated Interrupts

Interrupt commands enable the Supervisor to control I/O processing. These commands are constructed and sent by a TP under supervisory control to the IOP. They are then interpreted and executed in the IOP; thus, they may be considered commands.

The following paragraphs describe the basic procedure for executing an I/O process. Illustrated is the usage of some interrupt commands with major emphasis given to the point-in-time relationships of TP and IOP-channel functions.

A Start I/O interrupt is used to initiate an I/O process. To perform this operation, it is necessary for the programmer or Supervisor to:

1) Establish an I/O program in the command area.
2) Establish a list of descriptors in that part of the descriptor area assigned to the channel.
3) Load the channel name(s) and the address of the first command of the program into the START I/O command word.
4) Issue the START I/O Interrupt Command to IU, which in turn will direct it to the correct IOP.
5) Test the Interrupt Acknowledgment, IA, from the IOP for success or failure in initiating the new I/O process.

An '0' in the first flag of the Interrupt Acknowledgment indicates that the new I/O process has been successfully initiated. A '1' indicates failure, and the remainder of the IA should be examined to determine why the desired operation was not initiated.

Between the time a START I/O is issued by the TP , and the time the TP is released by the return of the Interrupt Acknowledgment, the IOP performs many functions:

1) It removes the Interrupt Command from the Exchange Net.

2) It decodes the command.

3) In the case of START I/O, it checks to see if the requested channel(s) are able to start a new process. If so, it puts a SUCCESS flag in the IA, and sends it to the requesting TP via the Exchange Net and the Interrupt Unit.

If a START I/O results in a SUCCESS condition, the channel registers necessary for program execution are loaded from the command area and the descriptor area. A successful program start does not necessarily imply that the device will begin successfully. Actual channel operation begins when the channel attempts to select the device over the I/O interface by propagating the device address. This operation and possible further data-transfer operations continue independently of the TP until either an END-flag descriptor occurs in the I/O program, the TP issues a HALT interrupt, or an error occurs. After the termination of an I/O process, the IOP requests an End Status Interrupt of any available TP. This interrupt alerts the original program of the end of its I/O process. The status of individual I/O operations can be surveyed by checking the descriptor area. Then, if conditions are propitious, new I/O process or processes can be started on the freed channels.

3.1.1 End Status Interrupt and Interrupt Acknowledgment

The End Status and Interrupt Acknowledgment are parallel in format and complementary in function to the Start I/O Command information. SIO gives a pointer to the beginning of a command string and names the channels to use in that I/O program. The End Status and Interrupt Acknowledgment return to the TP the last command pointer used and the final states of each channel used in the I/O program.

As in SIO, the End Status and Interrupt Acknowledgment are both delivered through the Exchange Net via the Interrupt Unit. But the Interrupt Acknowledgment is returned to the initiating TP as soon as the Interrupt Command is decoded. An End Status Interrupt, on the other hand, is sent at the completion of a program. And since delivery of the End Status is not a high priority task, this status may be delayed in the IOP. For this reason, an otherwise free channel(s) may be unavailable (actually in the Interruption Pending State) if it has not yet returned End Status from the IOP.

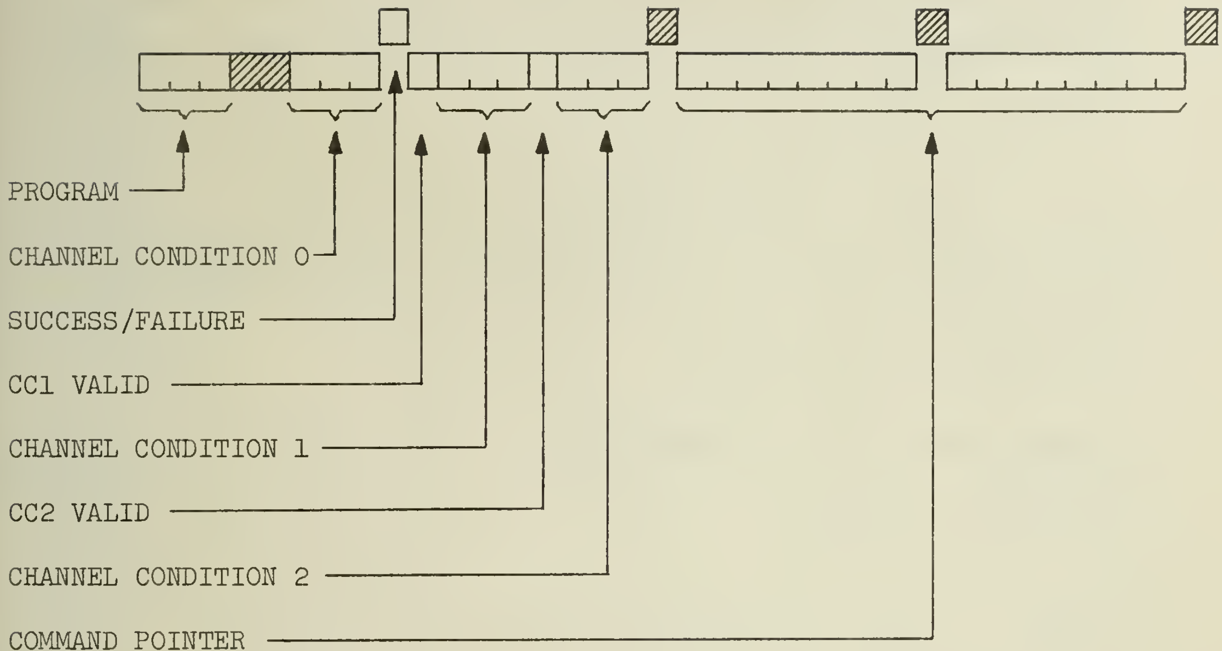


Figure 3.1.1 End Status and Interrupt Acknowledgment Format

The format is shown in Figure 3.1.1. The PROGRAM NAME (3 bits) is the same as the channel named in the CHO bits of the initiating Start I/O. CHANNEL CONDITION O (CCO) is the condition code (3 bits) associated with the channel named in the CHO bits of the initiating SIO. Likewise, CHANNEL CONDITION 1 (CC1) and CHANNEL CONDITION 2 (CC2) are associated with CH1 and CH2, respectively. A 'O' in the CC Valid bit means that there is no valid condition code for that channel. The Command Pointer contains the address of the last command loaded into the IOP from memory. The Command Pointer may be incorrect only if the last command executed modified the CP before failure.

CHANNEL CONDITION	ABBREVIATION	CODE	DEFINITION
Available	A	0	None of the following
Interruption Pending	I	1	End Status available from channel
Working	W	2	Channel operating on a program
Not Operational	N	3	Channel not operational

Table 3.1.1 Channel Condition Code

The success/failure flag is reset to 'O' at the completion of every successful interrupt command. It is set to '1' if the IOP is not able to complete the command and additional indicators describe the reason. If the command code is not valid, only the success/failure flag is set; the remainder of the IA is reset.

The success/failure flag is not valid for an End Status Interrupt.

3.1.2 Interrupt Command Format

The general format of an interrupt command is shown in Figure 3.1.2. Interrupt commands are always one word in length. The interpretation of the bit fields depends upon the command in question.

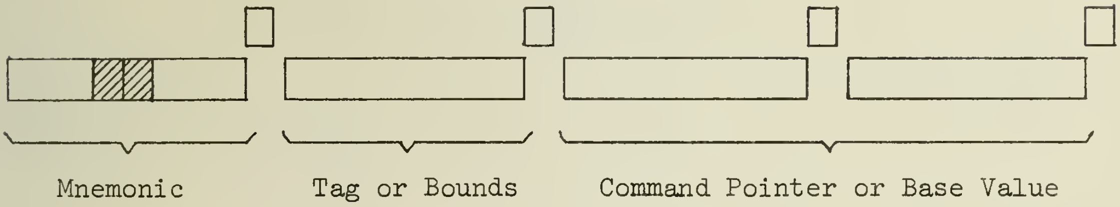
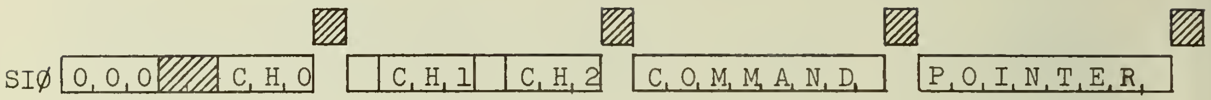


Figure 3.1. 2 Format of an Interrupt Command

3.1.3 Interrupt Commands

3.1.3.1 Start I/O



The interpretation of the SIO format includes the following definitions:

Bits 1-3:	COMMAND CODE:	000
Bits 4-5:	not used	
Bits 6-8:	CHO:	Designates one of eight channels. By convention CHO is also the program name.
Bit 9: flag	not used	
Bit 10:	CHANNEL VALID:	A '1' in this bit indicates that the following channel name is valid.
Bits 11-13:	CHANNEL:	These bits name a channel
Bit 14:	CHANNEL VALID:	A '1' in this bit indicates that the following channel name is valid.
Bits 15-17:	CHANNEL:	These bits name a channel
Bit 18: flag	not used	

If two CHANNEL locations are used, then by convention, these channel names should be stored in ascending order starting with CHO.

Bits 19-26, 28-35:	COMMAND POINTER:	Designates the location of the first command in the I/O program.
Bits 27, 36: flags	not used	

The execution of the SIO command includes the following steps:

1) Determine whether the named channel or channels are available. If any are not available, set the success/failure flag in the Interrupt Acknowledgment to '1', indicate which channels* are not available and send the Interrupt Acknowledgment word. Otherwise:

2) Load the program name, CHO, into the Key or Keys. Set each named channel RESERVED.

3) Load the second halfword from the Interrupt Command into the Command Pointer Register.

4) Set the Active indicator to '1'.

5) Reset the Interrupt Acknowledgment and send it to the Interrupt Unit.

6) Compute first descriptor location and load it into the Descriptor Pointer Register.

*The Condition Code of any channel which is not available is placed in the IA in bit positions parallel to those in the SIO command format. The Interrupt Acknowledgment and Channel Condition Code are described in Section 3.1.1.

3.1.3.2 Halt IØ

▨

HIØ

0	0	1	▨	PGM

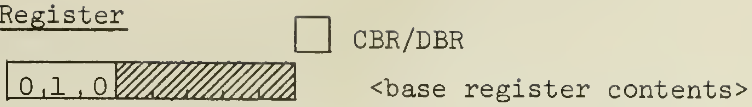
This command causes the IOP to terminate execution of the program identified by PGM. The Active indicator in the program status is reset to '0', thus removing this program from the IOP task list. Any working device is signaled to HALT, and all the channel 'Reserved' indicators are reset to '0' to release the channels.

The Interrupt Acknowledgment will contain the state of the program just before the Halt I/O is performed. The Channel-end Status Descriptor, if any, and the Program-end Status Descriptor and the End Status Interrupt will describe the state of the program after the Halt I/O is complete.

If the program ends before the Halt I/O arrives, the IA would be the only information generated by the HIO.

3.1.3.3 Load IØP Base Register

LIBR



This command loads the indicated base register from the rightmost three bytes and flags of the Interrupt Command. A'0' in the first flag position indicates the Command Base Register; a '1' indicates the Descriptor Base Register.

The operation of this command is also part of the startup procedure.

LIBR will not succeed if there is an active program in the IOP. Instead the Failure flag will be set, and the name of the highest priority program which is active will be placed in the PROGRAM bits of the Interrupt Acknowledgment.

3.2 Device Dependent Commands

The seven Device Dependent Commands, DDC's, are READ, READ REVERSE, SENSE, WRITE, CONTROL, TEST DEVICE STATUS, and POLL. An I/O program may consist of one or more DDC's and perhaps one or more Device Independent Commands, DIC's. The DDC's control communications with the I/O devices, and the DIC's, if needed, perform programming and "bookkeeping" tasks in the IOP and main storage.

In order to execute a program in the IOP, it is necessary for the programmer to:

1. Establish this program in the command area. (We assume that the base address is already in the Command Base Register in the IOP.)

2. Establish a descriptor list which corresponds to the command string. (We assume that the base address is already in the Descriptor Base Register.) Each DDC requires at least one Data Descriptor and space for the Channel-end Status Descriptor in the list. (The IOP maintains the space for the Program-end Status Descriptor.)

3. Create a START I/O Command which will direct the IOP to the areas established in steps 1 and 2. (The channel-program name in SIO directs the IOP to the first cell in the descriptor list, and the location of the first command in the program is found in the last half-word of SIO.)

4. Request that the supervisor issue the START I/O when the required channels become available. TP program control will then return to the original process. When the I/O process ends, the originating process will be notified via the Interrupt Unit. If the process fails during the Start I/O only the Interrupt Acknowledgment will be returned via the IU. If it ends any time after completion of the Start I/O, the End Status will be returned via the IU. Also in this later case, the Program-end Status Descriptor and the individual Channel-end Status Descriptors will be delivered via memory.

3.2.1 Format

Figure 3.2.1 shows the format of a Device Dependent Command (DDC). A DDC is distinguished from a Device Independent Command (DIC) by a zero flag bit over a mnemonic byte.

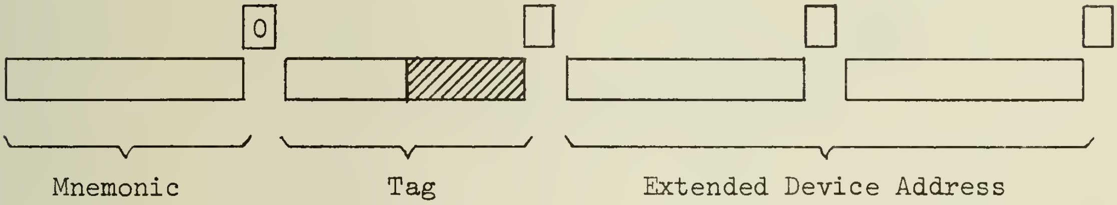


Figure 3.2.1 DDC Format

3.2.1.1 DDC Mnemonic Byte

The mnemonic byte specifies the basic I/O channel and device operation by use of an extended version of the IBM System/360 command code. See Table 3.2.1.1.

command	code
READ	MMMMMM10
READ REVERSE	MMMM1100
SENSE	MMMM0100
WRITE	MMMMMM01
CONTROL	MMMMMM11
TESTDS	MMMM1000
POLL	MMMM0000

Table 3.2.1.1 Command Code

M identifies a device dependent modifier bit. For example, the order to an IBM 1443 printer to "write and double space" is 00010001.

The POLL command is never sent to the device. It is executed entirely within the IOP and CIU.

3.2.1.2 DDC Tag Byte

The tag byte contains the command variant bits which describe the basic operation. The data descriptor also contains variant flags which further describe the operation.

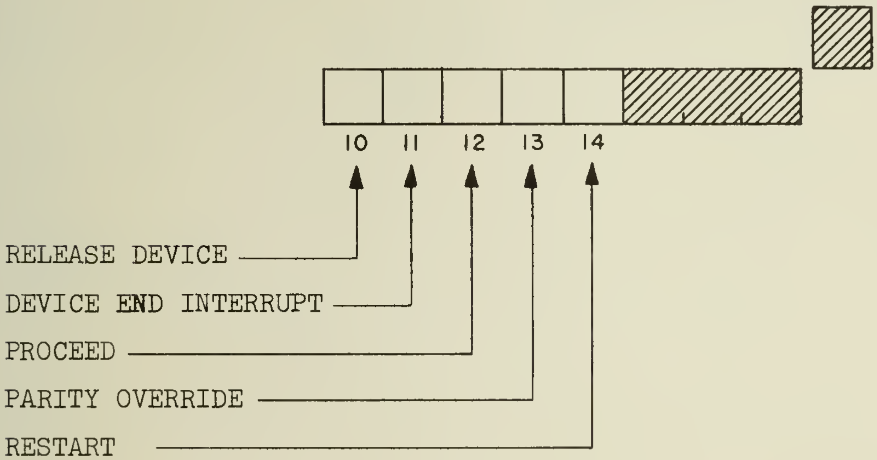


Figure 3.2.1.2 DDC Tag Byte

The command variant bits are defined below:

Bit 10: RELEASE DEVICE: A '1' indicates that the device should be released as soon as it sends "channel-end" status. A '0' indicates that the channel must wait for the "device-end" status.

Bit 11: DEVICE END INTERRUPT: A '1' indicates that the program should be interrupted if the device signals the end of transfer before the IOP does.

Bit 12: PROCEED: A '1' indicates that command chaining should occur. Depending on the I/O device, this may mean that certain device and device controller registers are not reset when a normal end of transmission occurs. At this point the IOP fetches and interprets the next I/O command and descriptor (which must be for the same device) without altering the Residual Capacity Register.

Bit 13: PARITY OVERRIDE: A '1' indicates the Parity Override variant. If a parity error is found in the IOP-CIU data path, the Data Parity Check indicator is set, but the data transfer continues norma

Bit 14: RESTART: A '1' indicates that the IOP must attempt to restart execution of a command which has been blocked by an initial device status report that contains a "Busy" indicator.

3.2.1.3 Extended Device Address

The extended device address, XDA, is composed of the device rate code, the channel to which the device is attached and the device address.

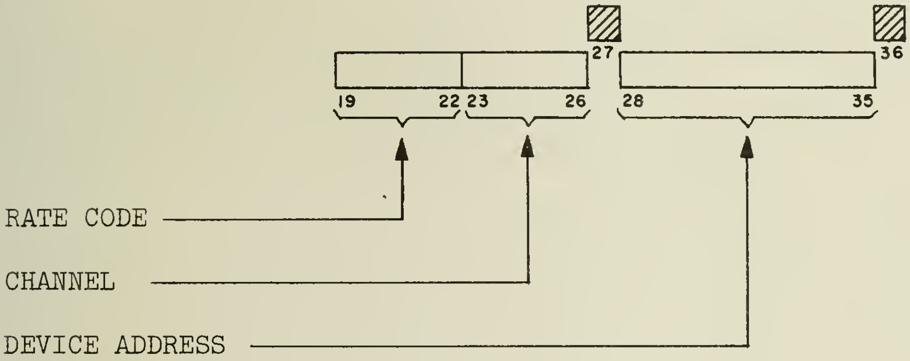


Figure 3.2.1.3 Extended Device Address Format

The parts of the extended device address are defined below:

Bits 19-22: RATE CODE: is related to the actual DEVICE RATE, DR, in bytes/sec. by $DR = 32 \times 2.^{(\text{RATE CODE})}$ The DEVICE RATE is compared with the Residual Capacity, RC. If DR is larger than RC, execution of the data transfer command is temporarily suppressed and the Capacity Limited Indicator is set until sufficient transfer capacity becomes available. If the Capacity Limited Interrupt flag (bit 27 in the data descriptor) is set and the IOP capacity proves insufficient, a Program-End Interrupt is generated instead of command execution merely being temporarily suppressed.

Bits 23-26: CHANNEL: Contains the name of one of the 16 Illiac III channels. The Key Register of the named channel is compared with the program register name. If they are not the same, the Channel Protect indicator is set and a Program-End Interrupt is generated.

Bit 27 flag: not used

Bits 28-35: DEVICE ADDRESS Contains the I/O Device Address DA which is sent over the X360 interface during the device selection sequence.

Bit 36: not used

3.2.2 I/O Descriptors

I/O descriptors are always a doubleword in length and they are stored in the descriptor segment within doubleword boundaries. The first halfword is always a link which points to the next descriptor in the list. The first two flags contain a code which indicates the type of descriptor, i.e. Data Descriptor or Status Descriptor.

Figure 3.2.2 illustrates the layout of the I/O descriptor segment. It is divided into four sections. Bytes 0-7 contain an Available Space Pointer which the Supervisor uses to keep an account of the available double words in the descriptor segment.

Bytes 8-71, the Program-end Status section, contain eight (the maximum number of concurrent programs) doubleword locations. Bytes 8-15 are assigned to program 0 for storage of one Program-end Status Descriptor. Consecutive doublewords are similarly assigned to consecutively numbered programs for storage of their final status.

Bytes 72-135, the Data Descriptor section, contain doubleword cells for storage of up to eight initial Data Descriptors for programs on channels 0-7 respectively. During the initiation of a new I/O program the location of the first data descriptor is computed (loc.= 8 (pgm +9)) and placed in the Descriptor Pointer Register for that program.

Starting with byte 136, the descriptor segment is divided into doubleword cells. The links in the first data descriptors (which are shown as arrows pointing out of the left side of the descriptor boundaries) are pointing to these cells which can be further linked together to form descriptor lists. These lists grow or shrink or even disappear as activity on the various program registers changes. But the location of the first cell in the descriptor list for a given program register is always found in a fixed location between byte 72 and byte 135.

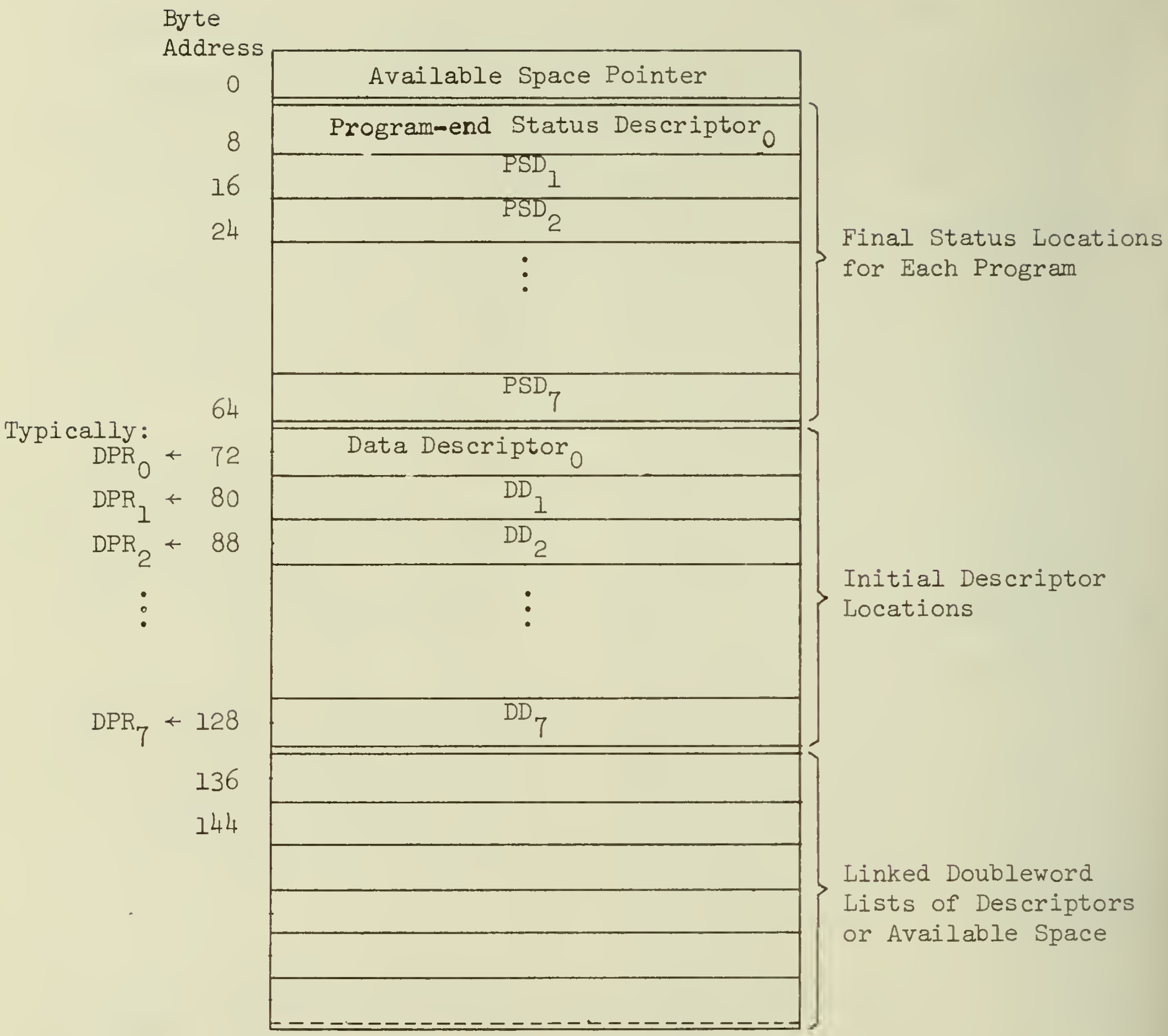


Figure 3.2.2 I/O Descriptor Segment

3.2.2.1 Data Descriptor

Figure 3.2.2.1 shows the general format of a data descriptor.

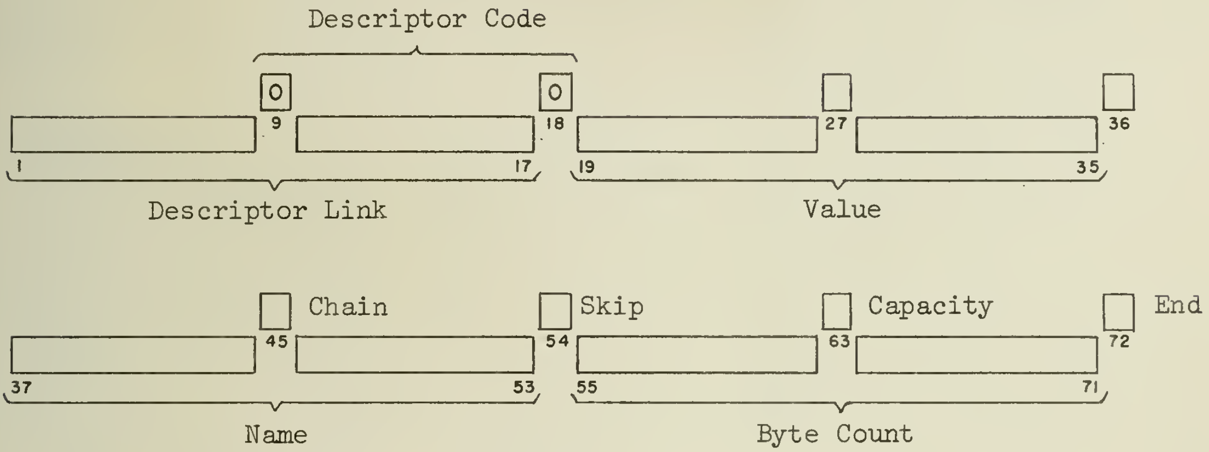


Figure 3.2.2.1 Data Descriptor Format

As soon as the IOP has determined that it has a valid DDC in the Command Register, it fetches a descriptor from the descriptor area in memory and loads the various registers. The first halfword of the descriptor is a link in the descriptor list. It is immediately written over the contents of the descriptor pointer register, DPR.

The various parts of the Data Descriptor have the following interpretations:

Bits 1-8, 10-17: DESCRIPTOR LINK: Contains the link to the next descriptor doubleword in the list.

Bits 9, 18 flags: DESCRIPTOR CODE: Contain the code which indicates the type of descriptor (in this case, 00, DD if previously unused; 11 if used.) Code contains a '00' when the descriptor list is first written. It is set to '11' by the IOP when this descriptor is fetched. The code flags permits the Supervisor to check on the progress of IOP execution.

Bits 19-36: VALUE: Pointer to data field in segment.

Bits 37-44, 46-53: NAME: Segment Name: address in Segment Table containing base information for the segment.

Bit 45 flag: CHAIN: A '1' indicates that data chaining should occur. A new Descriptor will be retrieved whenever the Residual Byte Count, RBC, is reduced to zero. Chaining will not proceed if the device signals the end of the present operati before the RBC is reduced to zero.

Bit 54: SKIP: A '1' indicates that data bytes are transferred from the device to the IOP, are counted, but are not placed in memory. (Note that this has no meaning for a writing-type command.)

Bits 19-26, 28-35: BYTE COUNT: Contain the number of bytes to be transferred in this I/O operation. (Byte Count must be zero if the programmer does not intend to send or receive any data other than device status.)

Bit 63 flag: CAPACITY: A '1' indicates that a Program-End Interrupt must be generated if the Capacity Limited Indicator turns on. See Rate Code, Section 3.2.1.3.

Bit 72: END: A '1' indicates that a Program-End Interrupt must be generated at the end of execution of this command.

3.2.2.2 Status Descriptors

I/O descriptor lists contain both data descriptors and status descriptors. The data descriptor describes the data operation to be performed by the IOP; the status descriptor describes the final state of the associated operations which the IOP has performed. The cells and links for both kinds of descriptors are set-up by the programmer who wishes to execute the I/O program or by the I/O Supervisor. But the information which is stored in the status descriptor is provided by the IOP at the completion of the operations. The IOP stores a Channel-end Status Descriptor, CSD, at the completion of every DDC and a Program-end Status Descriptor, PSD, at the end of every program.

3.2.2.2.1 Channel-end Status Descriptor

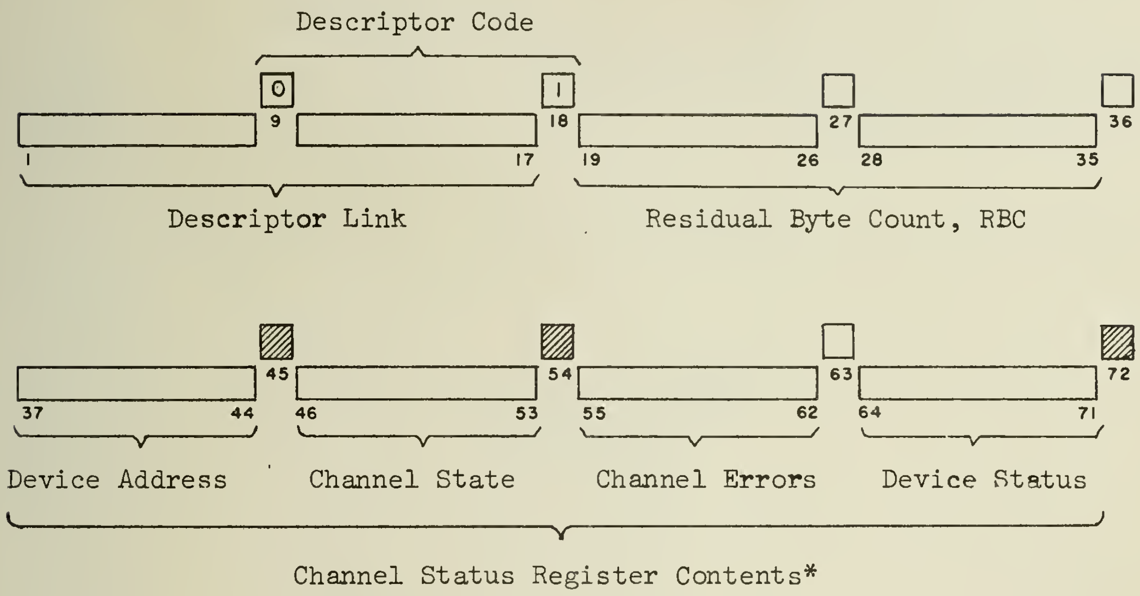


Figure 3.2.2.2.1 Channel-end Status Descriptor Format

At the end of every DDC, a Channel-end Status Descriptor is stored in the cell pointed to by the descriptor link retrieved with the previous data descriptor. The descriptor link, which is retrieved when the Channel-end Status Descriptor is stored, points to the Data Descriptor for the next DDC, unless the DDC ends in the middle of a data chaining operation.

* The first byte of the Channel Status Register, Channel Control, has been suppressed and replaced by the Device Address (See Section 3.2.1.3, bits 28-36).

The various parts of the CSD have the following interpretation:

Bits 1-8, 10-17:	DESCRIPTOR LINK:	Contain the link to the next descriptor doubleword in the descriptor list.
Bits 9,18 flags:	DESCRIPTOR CODE:	Contains the code which indicates the type of descriptor (in this case, 01, for CSD).
Bits 19-36:	RESIDUAL BYTE COUNT:	Contains the residual byte count from the BCR. When an input operation is terminated, the difference between the original byte count in the data descriptor and the residual byte count in the status descriptor is equal to the number of bytes transferred to main store; on an output operation, the difference is equal to the number transferred to the device.
Bits 37-45:	DEVICE ADDRESS:	Contains the Device Address which was sent over the X360 interface during the device selection sequence.
Bits 46-54:	CHANNEL STATE:	Contain the Channel State, which is stored from the Channel Status Register when the DDC is terminated.
Bits 55-63:	CHANNEL ERRORS:	Contains the Channel Errors, which is stored from the Channel Status Register when the DDC is terminated.
Bits 64-72:	DEVICE STATUS:	Contains the Device Status which the device controller sent over the X360 interface and the IOC stored in the Channel Status Register.

3.2.2.2.1.1 Channel State

Bits 46 through 54 of the Channel Status stored in the Channel-end Status Descriptor are called the Channel State. They are named after the corresponding byte of the Channel Status Register in which they are stored. They name the conditions which may exist when a DDC ends. See Figure 3.2.2.2.1.1 below for the format of this byte.

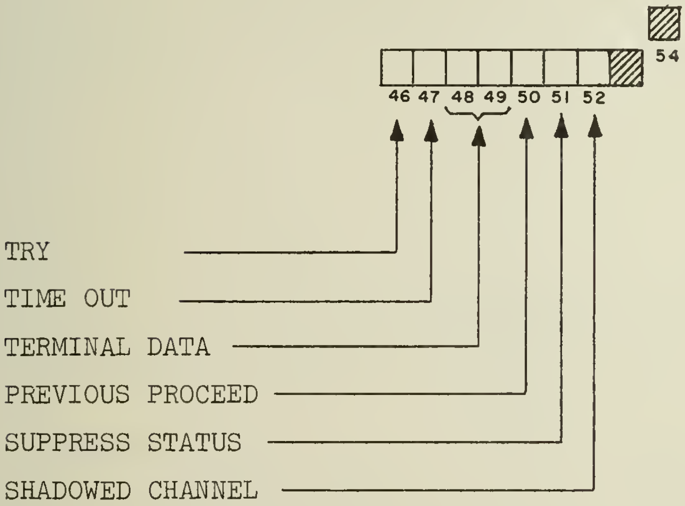


Figure 3.2.2.2.1.1 Channel State Format

The bits of the Channel State Byte are defined below:

Bit 46: TRY: A '1' indicates that the IOP is trying for the second time to get clear status from the device. (The programmer must have set the RESTART bit.)

Bit 47: TIME OUT: A '1' indicates that the IOP is waiting for a response from the device.

Bits 48-49: TERMINAL DATA: These bits indicate the stage of the termination process. '00' means that termination has not yet begun. '01' means that the last doubleword of data has been counted and the CIU is being notified. '10' means that the last doubleword of data is being transferred. '11' means that no further data transfers are expected. For further details on TERMINAL DATA especially in the case of CHAIN, see the Channel Subprocessor Flow Charts.

Bit 50: PREVIOUS PROCEED: A '1' indicates that the previous command started a 'command chain' or PROCEED.

Bit 51: SUPPRESS STATUS: A '1' indicates that the Channel-end Status Descriptor can not be stored because of a fatal error in the descriptor list.

Bit 52: SHADOWED CHANNEL: A '1' indicates that this channel is using the Shadow Descriptor Register.

3.2.2.2.1.2 Channel Errors

Bits 55 through 63 of the Channel Status stored in the Channel-end Status Descriptor are called the Channel Errors. They are named after the corresponding byte of the Channel Status Register in which they are stored. These bits might well be called the "Channel-end Status" since they name many of the channel conditions which may cause an operation to end and are not necessarily errors. For instance a DEVICE REQUESTED ending may be exactly what the I/O programmer intended. See Figure 3.2.2.2.1.2 below for the format of this byte.

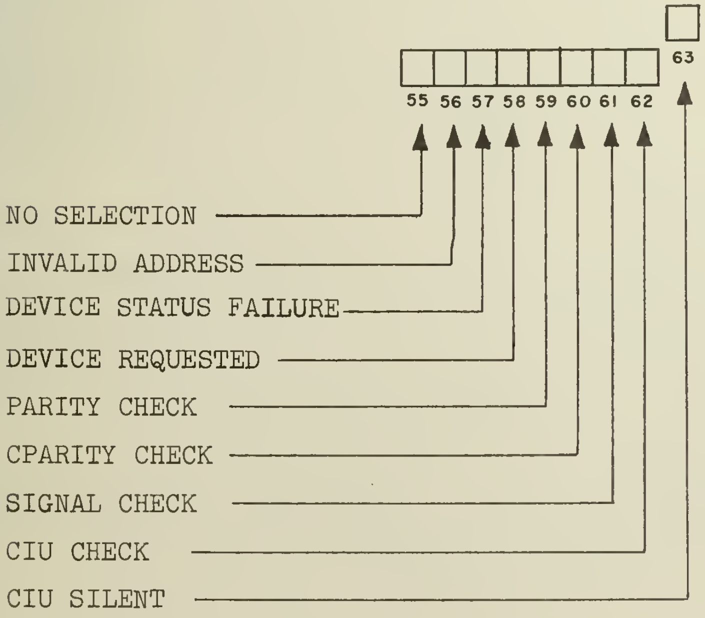


Figure 3.2.2.2.1.2 Channel Errors Format

The bits of the Channel Errors Byte are defined below:

Bit 55: NO SELECTION: A '1' indicates that no device has responded to the selection procedure or no devices were "requesting" during a POLL sequence.

Bit 56: INVALID ADDRESS: A '1' indicates that a device with the wrong address has responded to the selection procedure.

Bit 57: DEVICE STATUS FAIL: A '1' indicates that the device has presented an unacceptable device status byte.

Bit 58: DEVICE REQUESTED: A '1' indicates that the device requested more data in a WRITE operation or requested a transfer of more data in a READ operation after the IOP had reduced the Byte Count to zero and signalled the CIU to stop.

Bit 59: PARITY CHECK: A '1' indicates that the CIU sees a parity error on a data bus.

Bit 60: CPARITY CHECK: A '1' indicates that the CIU sees a parity error on the control lines, such as ADROUT, ADRIN, or CMDOUT.

Bit 61: SIGNAL CHECK: A '1' indicates an incorrect signal or combination of signals on the CIU/IOP interface.

Bit 62: CIU CHECK: A '1' indicates that the CIU has found an unacceptable condition in the CIU or on its incoming lines.

Bit 63: CIU SILENT: A '1' indicates that the CIU hasn't responded to the IOP raising ROYTO$_c$.

3.2.2.2.1.3 Device Status

Bits 64-72 of the Channel Status contain an 8-bit byte describing the status of an I/O device and its device controller. The status byte has a standard format for all I/O devices and it is described in detail in IBM System/360 literature, e.g., in the System/360 Interface Manual. Figure 3.2.2.2.1.3 shows its general format.

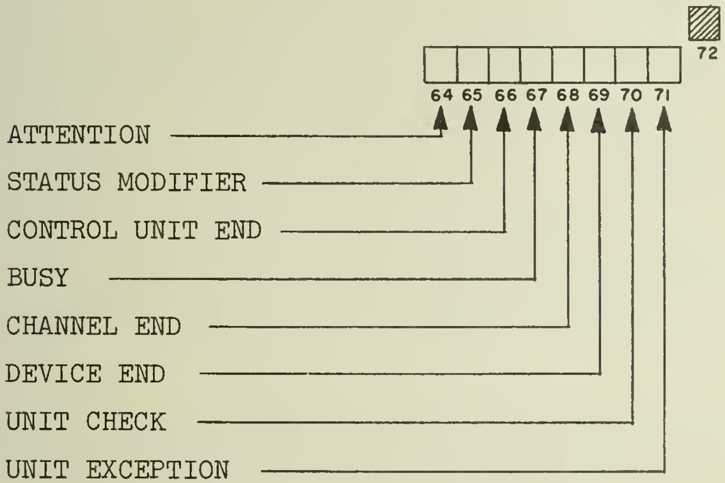


Figure 3.2.2.2.1.3 Device Status Byte

The device status byte is transmitted to the IOP (over the X/360 interface) in the following 5 situtations:

1) During the initial selection of the device by the IOP;
2) After termination of data transfer; the status byte is referred to as "channel-end" status in this case;
3) After completion of the entire operation at the device; the status byte is referred to as "device-end" status in this case;
4) To present unusual status conditions to the IOP;
5) To present any previously stored (or "stacked") status byte to the IOP.

The precise interpretation of device status byte depends on the device in question. The various bits have the following general interpretation.

bit 64: ATTENTION: This indicates that a timing error has occurred at the device.

bit 65: STATUS MODIFIER: This may indicate one of the following:

i) the device controller cannot provide current status,

ii) device controller busy as distinct from device busy,

iii) special ending condition.

bit 66: CONTROL UNIT END: This applies only to shared device controllers. It indicates either that the device controller was interrogated while busy or the device controller detected an unusual condition after channel-end status had been accepted.

bit 67: BUSY: This applies to an IOP-initiated selection sequence only and indicates that the device controller cannot execute the command.

bit 68: CHANNEL END: This marks the completion of data transfer.

bit 69: DEVICE END: This indicates that the I/O device has completed the current operation.

bit 70: UNIT CHECK: This indicates that the device or device controller has detected an unusual condition which is described in information available to the SENSE command, e.g., a programming error has occurred.

bit 71: UNIT EXCEPTION: This indicates detection of an unusual condition which is unique for a particular DDC, e.g., recognition of an end-of-tape mark. A SENSE operation is not required in response to a UNIT EXCEPTION condition.

bit 72: flag bit, not used.

3.2.2.2.2 Program-end Status Descriptor

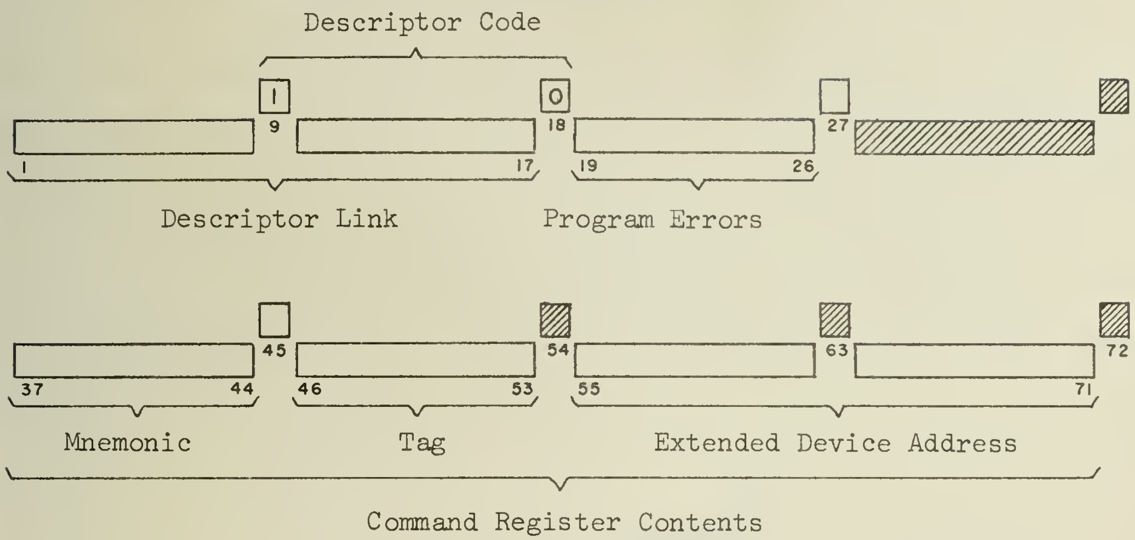


Figure 3.2.2.2.2 Program-end Status Descriptor Format

At the end of every I/O program, a Program-end Status Descriptor, PSD, is stored in the descriptor doubleword allotted to this program. The location is computed according to the following formula: loc = 8 (pgm + 1), where pgm is the number of the program register. When the PSD is stored, nothing is retrieved from main store, and no IOP registers are modified.

The various parts of the PSD have the following interpretations:

Bits 1-18: DESCRIPTOR LINK: Contains the last link and code which were stored in the Data Descriptor Pointer Register. If a DDC terminated normally at the end of the program, this link will point to the last status descriptor stored.

Bits 19-27: PROGRAM ERRORS: Contains the Program Errors portion of the Program Status Register at the end of the program.

Bits 28-36: not used

Bits 37-72: COMMAND REGISTER: Contains the Command Register contents.

3.2.2.2.2.1 Program Errors

Bits 19 through 27 of the Program-end Status Descriptor are called the Program Errors. They are named after the part of the Program Status Register in which they are stored. These bits might well be called "Program-end Status" since they name the IOP conditions which may cause a program to end and are not necessarily errors. For instance a CAPACITY LIMITED ending may be exactly what the I/O programmer expected. See Figure 3.2.2.2.2.1 below for the format of this byte.

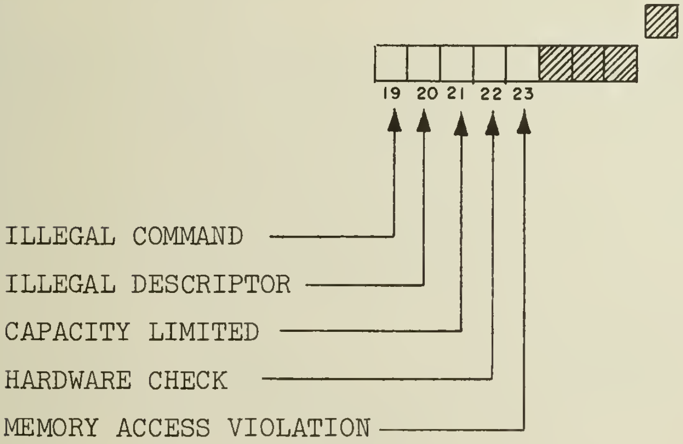


Figure 3.2.2.2.2.1 Program Errors Format

The bits of the Program Errors Byte are defined below:

Bit 19: ILLEGAL COMMAND: A '1' indicates a programming error in the DDC or DIC addressed by the Command Pointer.

Bit 20: ILLEGAL DESCRIPTOR: A '1' indicates a programming error in the Descriptor addressed by the Descriptor Pointer.

If both bits 19 and 20 contain '1's the DDC and the Descriptor are not compatible and the respective pointers address the error locations.

Bit 21: CAPACITY LIMITED: A '1' indicates that the DDC was attempting to operate a device with a data rate in excess of the available IOP capacity. This condition will cause a program-end only if the programmer sets the CAPACITY variant.

Bit 22: HARDWARE CHECK: A '1' indicates that the IOP detects a fatal hardware error in the IOP, XNET or MEMORY.

Bit 23: MEMORY VIOLATION: A '1' indicates that the program is attempting to access area outside of the allotted segments. If bit 19 is also '1', the Command Pointer is illegal. If bits 20 and 23 are '1' the Descriptor Pointer is illegal. If bit 23 alone is on, the Data Address indicates a non-existent location.

3.2.3 Basic Operations

The seven device-dependent commands READ, READ REVERSE, SENSE, WRITE, CONTROL, TEST DEVICE STATUS and POLL are individually described in the following sections. They may also be grouped and described in the three following functional subdivisions.

Reading-type command: a DDC which can cause data to be transferred from a device to main storage; i.e., READ, READ REVERSE, SENSE

Writing-type command: a DDC which can cause data to be transferred from main storage to a device; i.e. WRITE and CONTROL.

Reporting-type command: a DDC which initiates no action at the device, but which allows a device to report its status in a single status byte, i.e. TESTDS and POLL. Unlike the Reading or Writing-type commands, the Reporting-type commands never cause data to be transferred to or from main storage. Thus the device data rate fields in these commands are always ignored and not subtracted from Residual Capacity of the IOP.

3.2.3.1 Read

$\boxed{0}$

READ $\boxed{M\,M\,M\,M\,M\,M\,1\,0}$

READ causes the addressed device to initiate an input operation. Bytes are stored in main storage in an ascending sequence, beginning at the location (or cell) specified by the data address field of the Data Descriptor.

3.2.3.2 Read Reverse

$\boxed{0}$

READR $\boxed{M\,M\,M\,M\,1\,1\,0\,0}$

READR (for devices that can execute it) causes the addressed device to be started in reverse, and bytes to be transmitted to the IOP for storage in main storage in a descending sequence, beginning at the location specified by the data address field of the Data Descriptor. The order of the bytes within a doubleword is reversed by the CIU, but the bits within a byte are in the same order as when sent to the device on writing.

3.2.3.3 Sense

SENSE $\boxed{0}$

$\boxed{M\,M\,M\,M\,0\,1\,0\,0}$

SENSE can cause the addressed device to transmit one or more bytes of information, describing its current state. The bytes are stored in main storage in an ascending sequence, beginning with the memory location specified by the data address field in the Data Descriptor. The number of bytes transmitted and their meanings are a function of the modifier bits M, the device and its condition. SENSE can be used to obtain the current sector address from a disc or drum.

3.2.3.4 Write

$\boxed{0}$

WRITE $\boxed{M\ M\ M\ M\ M\ M\ 0\ 1}$

WRITE causes the addressed device to initiate an output operation. Bytes are read out in an ascending sequence starting from the memory location specified by the data address field of the Data Descriptor.

3.2.3.5 Control

$\boxed{0}$

CONTROL $\boxed{M\,M\,M\,M\,M\,M\,1\,1}$

Control is used to initiate special operations on the addressed device, and if necessary, transfer control data from ascending main storage locations to the device. For most control functions, the entire operation is specified by the modifier bits M in the comman code and the function is performed over the I/O interface as an immediate operation. If the command code does not specify the entire control function, the data address field in the Data Descriptor designates the location containing the required additional information.

3.2.3.6 Test Device Status

$\boxed{0}$

TESTDS $\boxed{\text{M M M M 1 0 0 0}}$

TESTDS causes the device controller to transfer one byte of status for the device designated in the command. This byte is stored in the device status field of the Channel Status Register. TESTDS does not cause data to be transferred to or from main storage.

3.2.3.7 Poll

$$\boxed{0}$$

POLL $\boxed{\mathrm{M\ M\ M\ M\ 0\ 0\ 0\ 0}}$

POLL causes the highest priority device (on the channel designated by this command) **with status** pending to transmit one byte of status and its device address. The status is stored in the device status field of the Channel Status Register and the device address is stored in the last byte of the Command Register. POLL does not cause data to be transferred to or from main storage.

3.3 Device Independent Commands

Device Independent Commands (DIC's) are executed by the IOP only and usually concern bookkeeping operations, e.g. address modification, loading and storing the contents of a program or channel register. DIC's do not directly initiate data transfer between an I/O device and main storage; accordingly no reference is made in these commands to a particular I/O device.

Execution of device independent commands is governed by the Task Subprocessor of the IOP; see Section 1.7.4.2.

3.3.1 Format and Interpretation

The format of a DIC is shown in Figure 3.3.1. The flag bit over the mnemonic byte is always set to '1' to indicate a DIC.

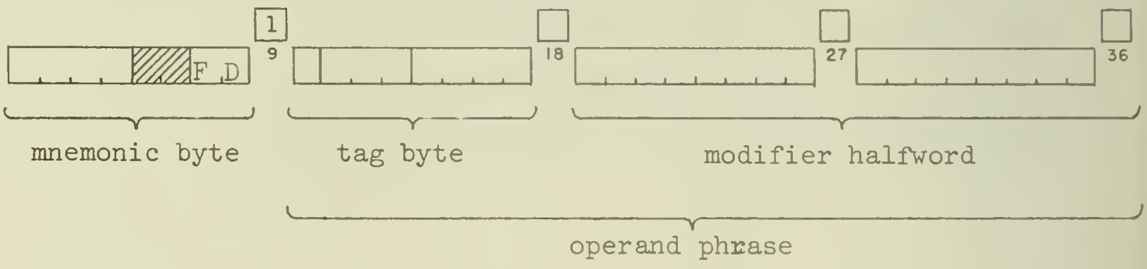


Figure 3.3.1 DIC Format

DIC's are always one word in length. The mnemonic byte consists of two parts: a command code and a cell size (FD).

The operand phrase consists of a tag byte followed by a modifier halfword. The tag byte specifies the number of the appropriate program/channel module and the (byte) address of the internal register to be operated upon. The modifier halfword consists of a 16-bit modifier value and two modification operator bits (the modifier flags) which indicate the use to be made of the modifier value.

3.3.1.1 DIC Mnemonic Byte

The Command Code (bits 1-4 of the mnemonic byte), specifies the basic operation to be executed by the IOP. The operation names are listed in Table 3.3.1.1.

Name	Code
LOAD	0001
STORE	0010
SETM	0011
RESETM	0100
TESTANY	0101
TESTALL	0110
MODIFY	0111
STØP	1000

Table 3.3.1.1 The Command Code

The Field Designator, FD, (bits 8-9 of the mnemonic byte) is interpreted as for the Taxicrinic Processor, namely

00	Byte
01	Halfword
10	Word (4 bytes)
11	Double Word (8 bytes)

3.3.1.2 DIC Tag Byte

The tag byte of a DIC specifies the (byte) address of an internal register within a program/channel module. First the choice of a program or channel module is indicated, followed by the 3-bit program/channel name (0-7). Finally the internal byte address (4 bits) of the register in question is identified. The field length (byte, halfword, word or doubleword) to be employed is specified in the FD bits of the mnemonic byte.

The format of the DIC tag byte is shown in Figure 3.3.1.2.

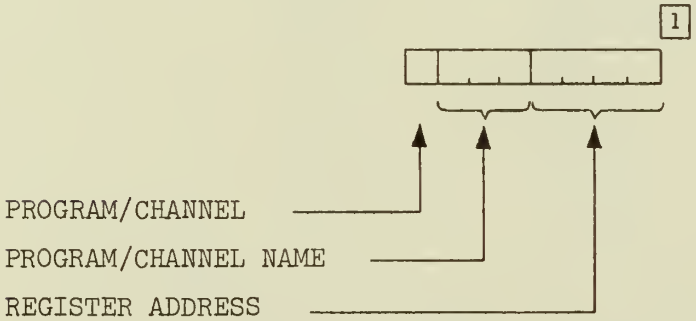


Figure 3.3.1.2 DIC Tag Byte

Register addresses for Program Modules run 0-7. The Command Register (CR) can be stored, but not loaded or otherwise modified by an input/output program.

Register addresses for Channel Modules run 0-12. The Shadow Descriptor Register (SDR) is not accessible to the programmer. Of course for memory protection neither base register: Channel Base Register (CBR) or Channel Shadow Base Register (CSBR) are accessible to the programmer.

3.3.1.3 Modifier Halfword

The rightmost halfword of a DIC contains a 16-bit modifier and two modification operation bits (flag bits 27 and 36). Aside from occasions when the main memory is being accessed, the modifier is treated as a non-negative integer (or mask), expanded appropriately with zeros to the left to fill the designated cell size. (For a designated cell size of "byte" only the rightmost byte of the modifier is used.

The modifier may be used in one of three ways as indicated by the modification operator bits. Table 3.3.1.3 lists these modification operations.

modification	symbol	code
REPLACEMENT	=	00
ADDITION	+	01
CONDITIONAL SUBTRACTION	-	10

Table 3.3.1.3 DIC Modification Operators

The modification operators are interpreted as follows:

1) REPLACEMENT: The modifier replaces the designated register.

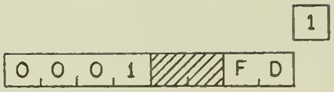
2) ADDITION: The modifier is added to the designated register; the resultant sum (modulo 2^{16}) is then used to replace the original register.

3) CONDITIONAL SUBTRACTION: The difference between the designated register value and the modifier is formed. If the difference is greater than zero, the difference replaces the register value and command execution proceeds normally. If the difference is less than or equal to zero, the register value is not modified, the command is not executed, and the next command is skipped.

3.3.2 Basic Operations

3.3.2.1 Load

LOAD



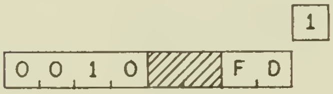
< operand>

The register specified by the DIC Tag Byte (more specifically, the portion of that register as specified by the FD) is loaded from the location specified by the operand phrase.

Indicators: access error, parity check,
bounds overflow

3.3.2.2 Store

STORE



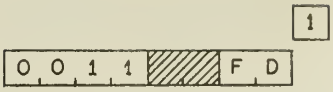
< operand>

The register specified by the DIC Tag Byte (more specifically, the portion of that register as specified by the FD) is loaded from the location specified by the operand phrase.

Indicators: access error, parity check,
bounds overflow

3.3.2.3 Set with Mask

SETM



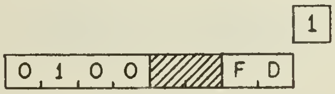
< operand>

A mask is used to selectively set those bits in the named register (more specifically, that portion of that register so specified by the FD) which are '1' in the mask. The location of the mask is specified by the operand phrase. Effectively, the mask is 'OR'ed with the specified register.

Indicators: access error, parity check,
bounds overflow

3.3.2.4 Reset with Mask

RESETM



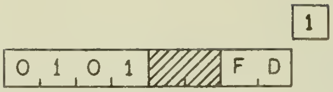
< operand>

A mask is used to selectively reset those bits in the named register (more specifically, that portion of that register as specified by the FD) which are '1' in the mask. The location of the mask is specified by the operand phrase. Effectively, the complement of the mask is 'AND'ed with specified register.

Indicators: access error, parity check, bounds
bounds overflow

3.3.2.5 Test for Any

TESTANY



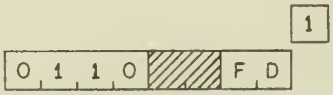
< operand>

The register specified by the DIC Tag Byte (more specifically, the portion of that register as specified by the FD) is compared with the mask. If any '1' bit in the mask is '1' in the named register, the test succeeds and the next command is executed. If the test fails, the next command is skipped. The location of the mask is specified by the Operand Phrase.

Indicators: access error, parity check,
bounds overflow

3.3.2.6 Test for All

TESTALL



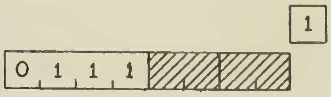
< operand>

The register specified by the DIC Tag Byte (more specifically, the portion of that register as specified by the FD) is compared with the mask. If all '1' bits in the mask are '1' in the named register, the test succeeds and the next command is executed. If the test fails, the next command is skipped. The location of the mask is specified by the Operand Phrase.

Indicators: access error, parity check,
bounds overflow

3.3.2.7 Modify Pointer

MODIFY



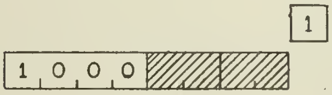
< operand>

Only a pointer modification, if indicated, will be performed. MODIFY can also serve as a NO-OP command for a suitable choice of modifier. A jump, equivalent to a TRANSFER IN CHANNEL (IBM), can be performed by doing a REPLACEMENT or ADDITION modification on the Command Pointer Register.

Indicators: access error, parity check,
bounds overflow

3.3.2.8 Stop Program

STOP



< operand>

The ACTIVE bit in the Program Status Register is reset to '0', thus removing this program from the IOP task list. All RESERVED bits in all associated Channel Status Registers are reset to '0'; thus removing these channels. INTERRUPT is set to call the TP's attention.

Indicators: program active, channel reserved

3.4 Peripheral Device Orders

This section differs from the pattern of other sections of this manual in that the diversity of available peripheral devices does not permit a closed internally-consistent treatment. In practice the universe of possible device attachments is limited by these constraints:

1) Data rates must be sufficiently high and the number of consecutive bytes transferred must be sufficiently long to warrant the channel overhead to establish communication. For example, teletypes and similar low speed terminal devices, in the Illiac III System are serviced by an independent mini-computer.

2) Attachment costs must not swamp the cost of the original equipment. Attachment requires providing a Device Controller to standardize channel communication. Much OEM equipment now has the Device Controllers built in--an advantage of the IBM compatible channel design.

3) Data formats must be compatible with the Extended IBM Interface (8 data bits and a flag bit). In many instances adequate compatibility is assured by simply suppressing the flag bit when communicating with other commercial computers.

4) Transmission distances must be consistent with the reply-back sequencing strategy used in the actual data transfers. For coupling to extensive high speed communication nets of more than 1000' a scheme of pseudo-replyback generation can be used, provided only synchronous data transmission is employed.

In this section, principles of channel operation are first introduced (Section 3.4), followed by the discussion of the CSP/IOP Interface in Section 3.5.

3.4.1 General Channel Operations

This section defines the overall purpose of each channel sequence and introduces terms used in the discussion of channel operations. The descriptions of the sequences are given only in the most general sense; specific details and exceptions are delegated to the CIU Manual.

The normal train of events on an I/O channel is the selection sequence, the data transfer sequence, and the ending sequence. The selection sequence connects the channel to a device, the data transfer sequence handles the movement of data between memory and the device, and the ending sequence collects ending status and clears the channel for future use.

3.4.1.1 Definitions

Device - dependent commands (DDC's) are classified in two different ways: according to function and according to device selection. The functional subdivisions are reading, writing, and reporting. Definition of these three follow.

Reading-type command: a DDC which can cause data to be transferred from a device to core storage; i.e., READ, READ REVERSE, and SENSE.

Writing-type command: a DDC which can cause data to be transferred from core storage to a device; i.e., WRITE and CONTROL.

Reporting-type command: a DDC which initiates no action at the device, but which allows a device to report its status in a single status byte, i.e., TESTDS and POLL.

The subdivisions according to the mode of device selection are addressing and polling. The specific definitions follow.

Addressing-type command: a DDC which has a data descriptor (hence a particular device address is specified); i.e., READ, READ REVERSE, WRITE, SENSE, CONTROL and TESTDS.

Polling-type command: a DDC which has no data descriptor (hence, no device address is given); i.e., POLL.

One further subgroup of DDC's is distinguished from the other DDC's due to certain special channel operations: the immediate-type command.

Immediate-type command: a DDC which

1. causes no transfer of data between core storage and the device, and
2. causes Channel End Status to be presented with the initial-status byte.

Control signals and data lines are classified according to the direction of information flow with respect to the Channel Subprocessor (CSP).

Incoming control signals: control signals which are directed toward the CSP; i.e., control signals from the Channel Interface Unit (CIU) to the CSP or from the Device Controller (DC) to the CIU.

Outgoing control signals: control signals which are directed away from the CSP; i.e., control signals from the CSP to the CIU or from the CIU to the DC.

Incoming channel data lines: channel data lines which are directed toward the CSP; i.e., data lines from the CIU to the CSP or from the DC to the CIU.

Outgoing channel data lines: channel data lines which are directed away from the CSP; i.e., data lines from the CSP to the CIU or from the CIU to the DC.

3.4.1.2 Normal Channel Conditions

Most outgoing control signals must be down before any sequences can be started. (See the section on Special Considerations in the CIU Manual for a discussion of the exceptions.) In like manner, all incoming control signals, except request in and metering in, must be down to successfully start a selection sequence. (Refer to appropriate section in the description of the device controller for details about the request in and metering in exceptions.)

3.4.1.3 Selection Sequence

The purpose of the selection sequence is to establish a communication link between the channel and a particular device. If the link is successfully initiated, the device presents status information to the channel.

The method of selection depends upon the type of device dependent command (DDC) being executed by the Channel Subprocessor (CSP). An addressing-type command results in an addressing selection sequence while a polling-type command starts a polling selection sequence. The details of each of these sequences are discussed in the following two sections.

The results of a selection sequence is either selection or no selection. If selection occurs, the selected device will send its status to the CSP. A polling selection sequence will terminate at this point, while an addressing selection sequence will either continue to the data transfer sequence or terminate, depending upon the information contained in the device status. If no selection occurs, the channel is cleared of outgoing control signals and the sequence is terminated.

3.4.1.3.1 Addressing Selection Sequence

In the addressing selection sequence the Channel Subprocessor (CSP) obtains an address of a specific device from the initial data descriptor and places it on the outgoing channel data lines. The device on the channel can respond by either placing its address or status on the incoming channel data lines or the device may not respond at all.

When the response is an address on the incoming channel data lines, the CSP checks this address against the device address sent. A correct match causes the CSP to place the actual command from the command register on the outgoing data lines. The device responds to this by sending its initial status back to the CSP on the incoming channel data lines.

Had an incorrect address match occurred, or if incorrect parity had been detected on the addresses or the command, the CSP would have cleared the channel control and data lines and tried the selection sequence again. A second failure will terminate the sequence.

If the initial response is status information, the associated device controller (DC) is busy, so the CSP accepts the status and terminates the sequence.

If no response is obtained by the CSP after a device address has been placed on the outgoing channel data lines, no selection results and the sequence is terminated.

3.4.1.3.2 Polling Selection Sequence

Devices are able to request service from the channel. The need for this ability arises in two cases in Illiac III:

1. When the channel disconnects from the device before it is able to present its ending status, and
2. When some asynchronous condition occurs in the I/O device which requires the channel's attention.

In both cases, the device needs to present status to the channel.

These service requests are handled by the polling selection sequence. (This sequence is known as the control-unit-initiated sequence on an IBM channel.) This sequence differs from the addressing selection sequence in two ways. The Channel Subprocessor (CSP) sends neither an address nor a command on the outgoing channel data lines. This means that the device controller (DC) will tell the CSP which device wishes access to the channel and that the DC will initiate any subsequent information transfer on the channel data lines.

The sequence proceeds as follows. The CSP starts up the channel but does not place an address on the outgoing channel data lines. Of the devices which require service, that device with the highest priority will reply by placing its address on the incoming channel data lines. The CSP will in turn acknowledge this address and then wait for the DC to initiate some information transfer sequence. In Illiac III the DC will present the device status, and then the sequence will terminate.

If no devices require service, no device address will be placed on the incoming channel data lines. In this case, the CSP will terminate the sequence.

3.4.1.4 Data Transfer Sequence

The purpose of the data transfer sequence is to transfer data between memory and the device. In the Illiac III system two transfer phases are required: (i) between the Channel Subprocessor (CSP) and the Channel Interface Unit (CIU), and (ii) between the CIU and the Device Controller (DC). This latter transfer phase between the CIU and the DC is approximately the same as the data transfer operations on an IBM subchannel.

3.4.1.4.1 Data Transfer Between CSP and CIU

Each data transfer sequence between the Channel Subprocessor (CSP) and the Channel Interface Unit (CIU) involves a double word (eight bytes) of data plus the parameters for controlling the data transfer between the CIU and the Device Controller (DC).

The CIU is in control of the data transfer portion of the sequence. The data transfer occurs as two transfers of four bytes each. The first transfer moves a word of data between the LR of the Permuter of the IOP and Buffer Register A of the CIU. The second transfer moves a word of data between the DR of the Permuter and Buffer Register B of the CIU. All of the data transferred need not be valid information; the number of valid bytes is given as part of the controlling parameters.

3.4.1.4.2 Data Transfer Between CIU and DC

Data transfer between the Channel Interface Unit (CIU), and the Device Controller (DC) is performed a byte at a time, with the DC in control with respect to initiating each transfer. Certain exceptional conditions, e.g., termination of byte count or parity error, will cause the CIU to stop the transfer.

When the transfer of bytes between the CIU and the DC satisfies the current byte count saved in the CIU, the CIU initiates a request for another data transfer between it and the Channel Subprocessor (CSP). Under ordinary conditions, the CIU will be able to continue the data transfer between it and the DC while waiting for CSP action, since the CIU has an additional word of data buffering. If the CSP does not reply to the CIU's request before the additional buffer is used up, the CIU will flag the CSP about this critical condition and further data transfer between the CIU and the DC will cease until a data transfer sequence is completed between the CSP and the CIU.

3.4.1.5 Ending Sequence

The purpose of the ending sequence is to close out a channel operation by gathering all ending information into the status register of the Channel Subprocessor (CSP) and by restoring the channel to the proper condition for future use.

Depending upon the conditions specified in the device dependent command (DDC) and the conditions on the channel, this sequence will do one or more of the following operations.

1. Collect and store the ending device status in the Channel Status Register.

2. Collect any remaining data and data byte count from the Channel Interface Unit (CIU), transmit the data to memory if appropriate, and correct the value of the residual byte count register in the CSP.

3. Set other status bits in the CSP which are necessary to properly indicate ending conditions.

4. Set the channel control lines to levels appropriate for the ending conditions.

3.5 CSP/CIU Interface

The principal uses of the interface between the Channel Subprocessor (CSP) and the Channel Interface Unit (CIU) are to convey control information associated with the initiation and termination of channel operations and to transfer data between the CSP and a CIU. This interface consists of a set of data and control lines which connect eight CIU's to their CSP. These lines are in two groups -- those which are shared in common by all eight CIU's and those which are separate for each CIU.

The shared lines are the two data buses, *bus to* and *bus from*, and the two control buses, *control to* and *control from*. The data buses carry all of the data between the CSP and the CIU's while the control buses regulate data flow and channel operations.

The separate lines are used to establish the connection between a particular CIU and the CSP. These lines are *ready to*, *ready from*, and *critical from*.

Figure 3.5 shows schematically the CSP/CIU interface. The interface signal names are named with respect to the CIU, that is, *to* signal names mean a signal *to* the CIU from the CSP while *from* signal names denote a signal originating *from* the CIU and going to the CSP. (The convention for the CIU/Device Controller (DC) interface is also with respect to the CIU, namely, *in* means signals coming *in* to the CIU from the DC and *out* means going *out* of the CIU to the DC.)

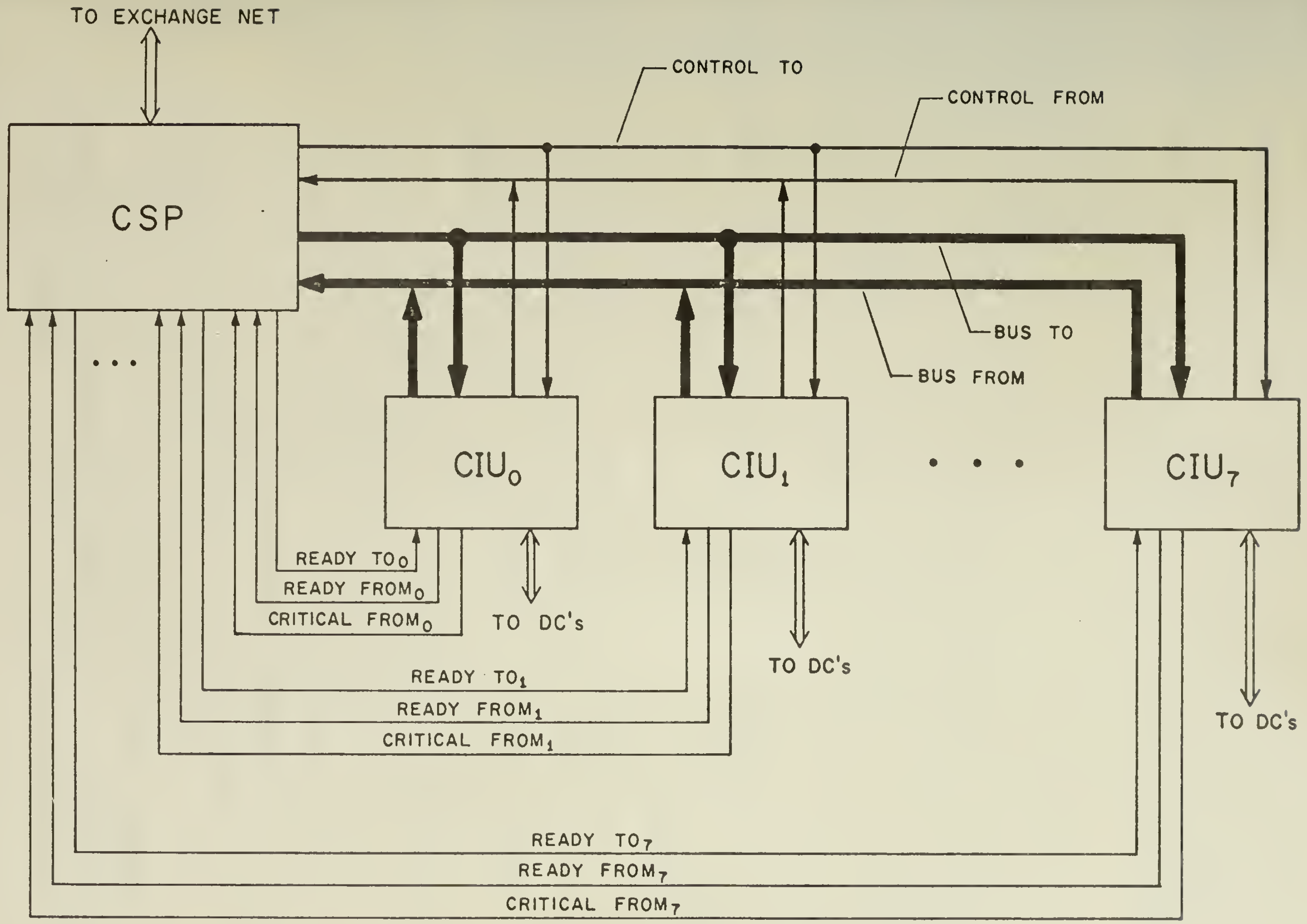


Figure 3.5 CSP/CIU Interface

3.5.1 Separate Lines

The separate lines between the Channel Subprocessor (CSP) and its eight Channel Interface Units (CIU's) are used as an overall control on the connection between the CSP and the CIU's. One ready to line runs from the CSP to each CIU while a ready from line and a critical from line run from each CIU to the CSP. (Refer to Figure 3.5.) The following is a brief description about how these lines are used.

When a CIU requires service from a CSP, it raises its ready from or its critical from line. The CSP replies by raising the ready to line to that CIU when it is able to handle this service request. The CIU in turn drops its ready from line (if it was up) and connects itself to the data and control buses of the CSP/CIU interface. The CIU will remain connected to the interface until the CSP drops ready to; then the CIU must disconnect itself from it. Thus the CSP controls the interface at all times.

The remainder of this section describes the separate lines in detail.

Ready To

The ready to lines are used by the CSP to connect and disconnect specific CIU's from the CSP/CIU interface. The CSP will raise the ready to line to a particular CIU either when it wishes to send some control information to that channel or when it is replying to a service request (ready from or critical from up). The former case occurs during execution of Start I/O, Halt I/O, program end, or start device dependent command (DDC). Here the CSP needs to transmit new control information to the channel, and so it initiates a channel connect sequence by raising ready to.

For the latter case (handling a channel's service request) the CSP will acknowledge the request only after it has handled all service requests from channels with higher priorities. Once this condition has been satisfied, the CSP will raise ready to to the requesting channel, wait for the CIU to connect to the CSP/CIU interface, and then proceed to handle the request based upon the current channel micro sequence (CMS) state and the information presented by the CIU on the incoming control signals and the incoming channel data lines.

After the CSP has transmitted the new control information or after the service request has been handled, it will drop ready to thus signalling that CIU to disconnect itself from the CSP/CIU interface.

Ready From

The CIU uses its ready from line to indicate to the CSP that some activity which requires the CSP's attention has taken place on the channel or in the CIU. In particular the following channel activities will cause the CIU to raise ready from. The raising of address in or select in by the Device Controller (DC) causes ready from to rise. The raising of status in by the DC will eventually cause ready from to rise: if a data transfer sequence between the CSP and the CIU is currently in progress, the rise in ready from is delayed until the data transfer sequence has been completed; otherwise, ready from is raised right after the status is put onto the bus from and status from is raised. Another channel condition which will cause ready from to rise is the fall of operational in while a disconnect sequence is in progress.

CIU conditions which cause the raising of ready from are the need for a CSP/CIU data transfer sequence and any CIU condition which causes the check from line to be raised. The CSP/CIU data transfer sequence is needed when the CIU satisfies the byte count in its byte count register during a reading-type command and when the CIU begins to transmit data to the DC from its C register during a writing-type command.

The CIU drops its ready from line as soon as ready to rises. After the ready from line has been so reset, it may rise again due to any of the above conditions even though ready to is still up.

Critical From

The purpose of the critical from line is to warn the CSP of an imminent overrun condition at the DC. The CIU raises its critical from line when the DC raises service in and either of the following is true: no valid byte count is in either the byte count buffer or the byte count register and the terminate device data flip-flop is off, or during a reading-type command the A, B, and C registers of the CIU are full.

The CIU will drop its critical from line either when the condition which caused it to be raised is rectified or when reset to is raised.

3.5.2 Data Buses

The purpose of the data buses is to provide word wide data paths for information flowing between memory [via the temporary storage register (TSR) of the I/O Processor (IOP)] and the three data buffer registers of each Channel Interface Unit (CIU). These word wide paths provide for four 9 bit bytes plus a bit of parity for each byte.

Bus To

Bus to provides a 40 bit data path from the output of the TSR permuter to the inputs of the A and B data buffer registers of all eight CIU's. Device addresses, device commands, and device data (for a writing-type command) are sent over this path a word at a time to the selected CIU.

Bus From

Bus from provides a 40 bit data path from the outputs of the A and B data buffer registers and the device status registers of the eight CIU's to the input of the TSR permuter. Device addresses, device status, device data (for a reading-type command), and channel ending conditions are sent over this path a word at a time from the selected CIU.

3.5.3 Control Buses

The purpose of the control buses is to provide regulating information on the channel. Each bus provides a 10 bit path between the Channel Subprocessor (CSP) and all eight Channel Interface Units (CIU's). The control to bus is the CSP's control link to the CIU's while the control from bus is the control link from the CIU's to the CSP.

3.5.3.1 Control To Bus

The control to bus is a 10 bit path from the Channel Subprocessor (CSP) to all the Channel Interface Units (CIU's). This bus is used to provide controlling information to the selected CIU from the CSP. The first 8 lines of the control to information can be interpreted either as a pilot byte or as an invoice byte. The pilot byte provides information to the CIU about how to set its outgoing control signals to the Device Controller (DC); the invoice byte provides the CIU with information it needs to control the flow of data between the CIU and the DC. Table 3.5.3.1 summarizes the names of the control to lines for the two different interpretations.

In addition to the pilot byte and the invoice byte the last two lines provide information on how to interpret the control to lines -- pilot to and monitor to.

Pilot To

The pilot to control line is used to indicate how lines 1 through 8 of the control to lines are to be interpreted by the CIU. When pilot to is up, lines 1 through 8 are to be handled as a pilot byte. When pilot to is down, the CIU handles lines 1 through 8 as an invoice byte. The CSP sets pilot to while monitor to is down to reflect the type of byte it wishes the CIU to decode the next time monitor to is up.

Monitor to

The monitor to control line is used as an interlock for the rest of the control to signals. When monitor to is down, the CIU is to ignore the condition of the other control to lines. When the CSP raises monitor to, the CIU is to monitor the control to lines; that is, it is to check for changes in the control to signals that have occurred since the last time monitor to was up (and take the appropriate actions for any changes that are noted) and to begin monitoring the control to lines for further changes.

LINE	NAME	ABBREVIATION	
1	Clock To	Clk To	Pilot Byte
2	Metering To	Mtr To	
3	Operational To	Opl To	
4	Address To	Adr To	
5	Select To	Sel To	
6	Command To	Cmd To	
7	Suppress To	Sup To	
8	Reset To	Rst To	
1	Count Valid To	CV To	Invoice Byte
2	Byte Count To 1	BC To 1	
3	Byte Count To 2	BC To 2	
4	Byte Count To 3	BC To 3	
5	Count End To	CE To	
6	Parity Override To	PO To	
7	Data To	Dta To	
8	No Data To	ND To	
9	Pilot To	Plt To	
10	Monitor To	Mon To	

Table 3.5.3.1 Control To Bus Signal Names

3.5.3.1.1 Pilot Byte

The pilot byte controls most of the Channel Interface Unit (CIU) control out lines and provides certain information for the CIU to control its non-data handling sequences. The names of the first seven lines of this byte correspond to names of seven of the control out lines of the CIU.

Clock To

The purpose of clock to is to control clock out which in turn provides an interlock for the Device Controllers (DC's). The Channel Subprocessor (CSP) raises clock to on all channels that are reserved during the execution of a Start I/O interrupt command. The addressed CIU's respond by raising their respective clock out lines.

The CSP drops clock to on all channels which were reserved for a given program after the program-end status descriptor (PSD) has been placed in the memory. The addressed CIU's respond by dropping their clock out lines.

Metering To

The purpose of metering to is to control metering out which in turn is used to condition all other meters in the DC's. Metering to is raised by the CSP on all channels which are reserved during the execution of a Start I/O interrupt command. The addressed CIU's respond by raising their respective metering out lines.

After the program-end status descriptor (PSD) has been stored in memory, the CSP drops metering to on all channels which had been reserved for that program. Metering out is then dropped by the respective CIU's.

Operational To

The purpose of operational to is to control operational out which in turn is used by the DC as an interlock on the other control out signals. The CSP raises operational to to all CIU's at the completion of the power on sequence. The CIU's respond by raising their respective operational to lines.

The only time that the CSP will change operational to is for a selective (malfunction) reset or for a system reset. For a selective reset, the CSP drops operational to while simultaneously raising suppress to. The CIU responds by raising suppress out, setting its disconnect sequence flip-flop, waiting 250 nanoseconds, and then dropping operational out. For a system reset the CSP will drop operational to (suppress to should already be down). The CIU responds by dropping operational out. For both resets the CSP will raise operational to after at least 6 microseconds have elapsed. The CIU then raises operational out.

Address To

Address to is used to partially control address out and certain sequences in the CIU. The usual situation occurs when the CSP raises address to in channel micro sequence (CMS) state 0 for an addressing-type command. Since this is associated with the raising of select to, the CIU interprets this as an initial selection sequence. Thus the CIU sets its initial selection sequence flag (ISSF) on, loads the information on bus to into its A register, gates the address portion of that information to the bus out, and sets address out up. (Since select to is up, the CIU will next raise hold out and then select out.

The CSP will drop address to in CMS state 1 for the above mentioned situation. If the CSP also drops select to at the same time, this means that the CIU must execute a disconnect (from the DC) sequence, since either no selection has taken place or the DC was busy. If no selection took place (select in is up), the CIU drops select out and hold out and then if the CIU has a device address on bus out, it removes this address from bus out and drops address out. If the DC was busy (status in is up), the CIU drops select out and hold out, waits until status in falls, and then removes the device address from bus out and drops address out.

If the CSP does not drop select to at the same time it drops address to, the CIU takes no action, since this implies a DC connect situation where the device address on bus out has already been removed and address out has already been dropped by the CIU in response to the DC's raising of operational in.

The CSP also raises address to to indicate an interface disconnect sequence. The CIU identifies this sequence by the fact that the CSP dropped select to at the same time it raised address to. The CIU responds to this by first raising address out, setting its disconnect sequence flip-flop, and then dropping hold out and select out. When the interface disconnect sequence is complete, the CSP will drop address to. To this the CIU responds by dropping address out.

Select To

The purpose of select to is to control select out and hold out and to partially control address out. The CSP raises select to in CMS state 0. If address to is also up, the CIU responds as described under the discussion about address to; otherwise, the CIU first raises hold out and then it raises select out.

The CSP will drop select to in CMS state 1, 2, or 3 depending upon the response of the DC and/or the type of command being executed. The response of the CIU in all cases is to drop hold out and select out.

Command To

Command to is used by the CSP to partially control the raising of command out and to direct certain CIU actions. The CSP raises command out in CMS state 1 after a valid address has been received from the DC via the CIU. The CIU responds by first removing the device address from bus from. Then if the initial selection sequence flag (ISSF) is on, the CIU interprets the command that is stored in the second byte of the A register in order to set its format flip-flops, puts the command onto bus out, and raises command out.

If the ISSF is off, the CIU puts a byte of all zeros onto bus out, and then it raises command out.

The CSP drops command to at the beginning of CMS state 2. The CIU takes no action, since it would have already dropped command out in response to the DC's dropping of address in.

Suppress To

The purpose of suppress to is to partially control suppress out. It is associated with the proceed variant (command chaining) of a device dependent command (DDC) and with selective reset. In the case of proceed the CSP will raise suppress to before accepting device status in CMS states 2 or 3. The CIU responds by raising suppress out. The CIU must insure that it does not subsequently raise service out for at least 250 nanoseconds since it raised suppress out.

Normally the CSP drops suppress to during CMS state 1. The CIU will take no action, since it should have already dropped suppress out when the DC raised operational in.

The CSP will also drop suppress to during CMS states 0 and 3 (assuming that suppress to is currently up due to a proceed condition) if certain conditions are detected that automatically terminate the proceed variant. Detection of a status error in CMS state 3 or the previous proceed flag going off in CMS state 0 are examples of these conditions. In these cases the CIU is to drop suppress out. A timing constraint applies in the status error condition: the CIU must insure that it does not raise service out until at least 250 nanoseconds have elapsed since the dropping of suppress out.

In the case of selective reset the CSP raises suppress to at the same time that it drops operational to. The CIU responds by raising suppress out, waiting 250 nanoseconds, and then dropping operational out. At the end of the reset time, the CSP will simultaneously raise operational to and drop suppress to. The CIU is to raise operational out, wait 250 nanoseconds, and then drop suppress out.

Reset To

The purpose of reset to is to put all the CIU registers and flags into their reset condition. Specifically, when the CSP raises reset to (this is normally done in CMS state 0), the CIU is to clear all of its registers to zero and turn off all gates associated with its registers, reset all of its status and condition flip-flops, set all from flip-flops to zero, and set all to flip-flops to agree with the control to lines. After the from flip-flops are set to zero, the control in lines which are up will again set their respective flip-flops on.

The CIU takes no action when reset to is dropped by the CSP.

3.5.3.1.2 Invoice Byte

The invoice byte provides the controlling information for handling data transfer both between the Channel Subprocessor (CSP) and the Channel Interface Unit (CIU) and between the CIU and the Device Controller (DC). The signal names of this byte reflect the main purpose of the signals; however, some of the signals such as no data to take on additional meaning when used with other signals.

Count Valid To

The purpose of count valid to is to indicate to the CIU when a valid byte count is on the byte count to lines. The CSP raises count valid to during channel micro sequence (CMS) states 2 and 3 when it sends a new byte count to the CIU. The CIU responds by gating the value of the byte count to lines into the byte count buffer if no data to is down. If the count is gated into the byte count buffer (BB), the CIU next gates this into the byte count register (BC) if the register is empty. Then regardless of how the byte count was handled, the CIU gates the value of the count end to line into the terminate device data flip-flop.

The CIU takes no action when count valid to is subsequently dropped by the CSP.

Byte Count To

The byte count to control lines are used to transmit byte count information from the CSP to the CIU. The information on these lines is valid only when count valid to is up. The discussion of count valid to describes the gating sequence.

When the byte count to lines are valid, they can take on values ranging from 000 to 111. The values from 001 to 111 are interpreted as the decimal equivalent of the binary numbers; that is, 001 is interpreted as 1 decimal, 010 is 2, 011 is 3, and so on up to 111 which is 7. The value 000 is defined to be 8. Therefore, a byte count of 0 cannot be communicated from the CSP to the CIU over the byte count to lines; such information must be passed by using the no data to line. Refer to the discussion of no data to line for further details.

Count End To

The purpose of count end to is to inform the CIU that a given byte count is the last one for the current channel operation. It is gated into the terminate device data flip-flop under control of the count valid to line.

During normal channel operations count end to will be down every time the CSP sends another byte count to the CIU. When the last count is put onto the byte count to control lines, the CSP also raises count end to. Then count valid to is raised to signal the CIU to load the new byte count and count end information. Subsequently the CIU sets its terminate device data flip-flop, since the count end to line was up. Eventually this last count is moved down to the byte count register (BC) where it is then used in the comparison with N (the CIU's byte counter value) each time the service in line falls. If the DC requests a sufficient number of data transfers between it and the CIU, this comparison of N with this last count in BC will yield equality, and the CIU will set the byte count register flip-flop (BCFF) off and initiate a data transfer cycle between it and the CSP. Since the last byte count has been transferred to the CIU, no further counts will be sent to the CIU, and hence BC and the byte count buffer (BB) will remain empty. If the DC attempts any further data transfers the CIU upon checking BCFF and byte count buffer flip-flop (BBFF) and finding them both off will then check the terminate device data flip-flop. Since this is on, the CIU will begin a stop sequence to the DC by raising command out. (The stop sequence is completed when the DC drops service in and the CIU drops command out.)

Parity Override To

The parity override to control line is used to determine the action that the CIU should take upon detecting a data parity error.

When the CSP raises *parity override to*, the CIU sets its parity override flip-flop. When the CSP drops *parity override to*, the CIU resets its parity override flip-flop.

The condition of this parity override flip-flop determines the CIU action to parity errors that it detects occurring on either the *bus out* or the *bus in* lines. When a parity error is detected on either of these buses while data is being transferred (note that this does not include addresses, commands, or status), the CIU will first set its data parity indicator flag. If the parity override flip-flop is on, the CIU then continues to handle the offending data byte as though nothing were wrong with it. If the parity override flip-flop is off, the CIU initiates a stop sequence on the bus with the parity error. (This means that the offending data byte will not be transferred to its destination.)

The data parity indicator flag will set *parity from* the next time that the CIU initiates a data transfer cycle between it and the CSP (that is, the next time *data from* is raised).

Data To

The *data to* control line is used to regulate the flow of data and status between the CIU and the CSP. The CSP raises *data to* in response to the CIU's raising of *data from* after the CSP determines that it is ready for an information transfer between it and the CIU. The CIU responds by dropping *data from* or *status from* after appropriately changing certain gates and indicators. Refer to the discussion on the information transfer cycle between the CSP and the CIU for full details. The CSP responds to the falling CIU control signal by dropping *data to*. If data (rather than status) was transferred in the above operation, a second cycle similar to the first involving *data from* and *data to* will occur to transmit the second data word.

If the first information transfer involved the initial or second device status byte, then after transmitting the status byte, the CIU will raise data from in order to start a data transfer cycle. The CSP responds by raising either data to (thus continuing the data transfer cycle) or no data to (thus stopping the data transfer cycle) depending upon the current conditions in the CSP.

No Data To

The purpose of no data to is to allow the CSP to suppress a CSP/CIU data transfer cycle and to conditionally control the transfer of the byte count. To suppress a CSP/CIU data transfer cycle the CSP will raise no data to in CMS states 2 or 3 (the CIU had previously raised data from to indicate that it expects a CSP/CIU data transfer cycle). The CIU responds by dropping data from, resetting its data transfer sequence (DTS) indicator, and resetting the byte count buffer flip-flop (BBFF) and the byte count register flip-flop (BCFF) to zero.

If in addition to stopping the data transfer sequence the CSP wishes to transmit byte count information to the CIU, two cases arise depending upon the value of the byte count to be sent. In the one case if a non-zero value is to be sent, the CSP first drops no data to before raising count valid to. (The CSP has, of course, placed a valid count on the byte count to lines. The CIU follows its normal response to this; that is, it loads the value of the byte count to lines into the byte count buffer.)

In the second case if a zero byte count is to be sent (which also implies the end of the count), the CSP leaves no data to up while raising count end to followed by count valid to. The CIU interprets this condition (count end to and no data to both up at the same time) to mean a zero byte count; thus it sets its terminate device data flip-flop but it does not load its byte count buffer.

In any case the CIU takes no further action when the CSP drops the no data to control signal.

3.5.3.2 Control From Bus

The control from bus is a 10 line path from all the Channel Interface Units (CIU's) to the Channel Subprocessor (CSP). This bus provides information about the operating status of the CIU and the condition of the incoming control signals from the Device Controller (DC). Table 3.5.3.2 summarizes the names of the control from lines. The first 6 lines reflect the conditions of corresponding control in bus lines from the DC to the CIU. The remaining 4 lines reflect control status of the CIU.

LINE	NAME	ABBREVIATION
1	Select From	Sel Fr
2	Operational From	Opl Fr
3	Address From	Adr Fr
4	Metering From	Mtr Fr
5	Status From	Sta Fr
6	Request From	Req Fr
7	Data From	Dta Fr
8	Parity From	Par Fr
9	Check From	Chk Fr
10	Reply From	Rpy Fr

Table 3.5.3.2 Control From Bus Signal Names

Select From, Operational From, Address From, Metering From, Status From, Request From

Select from, operational from, address from, metering from, status from, and request from reflect the condition of their respective control lines for the DC's to the CIU; namely, select in, operational in, address in, metering in, status in, and request in. Each of these from control lines will be up when its corresponding in control line is up, and it will be down when the corresponding in control line is down.

Data From

Data from is used by the CIU to regulate data transmission between it and the CSP. Right after transferring the initial device status to the CSP the CIU raises data from in order to initiate the transfer of the initial byte count and in the case of a writing-type command the first double word of data. For other than the initial data transfer the CIU will raise data from whenever its A and B buffer registers become full while executing a reading-type command or whenever its A and B buffer registers both become empty while executing a writing-type command.

The CIU will drop data from in response to the raising of either data to or no data to. For normal data transfers the CIU will raise data from a second time when it is ready to handle the transfer of the second word of data. The CIU then drops data from when data to is again raised by the CSP.

Parity From

Parity from is raised by the CIU to indicate that some sort of parity error has been detected in the CIU. The condition of the other control from lines will indicate the source of the parity error. For example, if parity from and address from are both up, this means that the CIU detected a parity error either on the outgoing device address or the incoming device address. Parity errors on data may stop the CIU from making further data transfers between it and the DC depending upon the condition of the parity override flip-flop.

The CIU drops parity from at the end of the next double word transfer between the CIU and the CSP for data parity errors. For other types of parity errors the CIU will drop parity from only when the CSP raises reset to.

Check From

Check from is raised by the CIU to indicate that some illegal condition was detected and that it is unable to continue its current sequence. The CIU will drop check from only in response to the CSP's raising of reset to.

Reply From

Reply from is used by the CIU in reply to ready to. It is raised by the CIU after it has connected itself to the CSP/CIU control and data buses in response to the raising of ready to. Reply from is dropped by the CIU immediately before it disconnects itself from the CSP/CIU control and data buses in response to the dropping of ready to. Thus this signal is used by the CSP primarily to determine if the addressed CIU is alive (that is, not dead).

3.6 Scan-Display System

The Scan-Display System of the Illiac III computer is described on pages 187-206 of the Proceedings of the 1969 Spring Joint Computer Conference and is included herein for completeness. An errata for this paper is given in section 3.6.1.

3.6.1 Errata to Scan-Display System

The following changes should be made to the paper "Parametric Description of a Scan-Display System" by L. A. Dunn, et. al., published in the Proceedings of the 1969 Spring Joint Computer Conference:

page 191, column 2 lines 23 and 24 should be:

$$\Delta X = 2^p \text{ basic cells} = 2^{p-(b_g+k)} \text{ units} \qquad (1)$$

$$\Delta Y = 2^q \text{ basic cells} = 2^{q-(b_g+k)} \text{ units} \qquad (2)$$

page 192, column 1 line 26 should be:

as 2^n, hence the number of possible states as 2^{2^n}. The

page 192, column 1 lines 32 to 38 should be:

The parameter B_s will distinguish between a spot and a slit. If the slit option is taken, then the parameters u and v define the sample width and length as 4^u and 2^v gross basic cells, respectively. If the spot option is taken, the circular spot size can be specified by u with v = 0. Thus the smallest spot size (u = 0, v = 0) is approximately 1 gross basic cell in diameter.

page 195, column 1 line 3 should be:

Since the video line sweep time, $\Delta t(i)$, is a fixed

page 196, column 2 lines 21 to 25 should be:

States	Source	Destination	Command	Figure
4	$(F \oplus V)$ ·	$(C \oplus \overline{C})$ ·	READ	15
1	F ·	$V \cdot \overline{C}$ ·	READ	16
1	V ·	$F \cdot \overline{C}$ ·	WRITE	16
4	C ·	$(F \oplus \overline{F}) \cdot (V \oplus \overline{V}) \cdot$	WRITE	17

page 205, column 1 lines 34 to 36 should be:

c_1 Sweep velocity constant (with $c_1 = 2^{n_{max}} \cdot f_c$, V_s is given in basic cells per microsecond)

3.6.2 Data Formats for Scan-Display System

Three forms of data strings are accepted as input by the S-M-V controller: Coordinate, Incremental, and Raster. Two forms are generated as output by the S-M-V controller: Coordinate and Raster. The formats for these strings are detailed in the next three sections.

3.6.2.1 Coordinate Format

The coordinate data string used by the S-M-V controller is similar to that used by the PAU. The format is as follows (assuming an X-axis scan):

$Y^{1}[1]\quad \theta[1,1]\quad X[1,1,1]\quad G[1,1,1]\ldots X[1,1,m]\quad G[1,1,m]$

$Y^{1}[1]\quad \theta[1,2]\quad X[1,2,1]\quad G[1,2,1]\ldots X[1,2,n]\quad G[1,2,n]$

.
.
.

$Y^{1}[1]\quad \theta[1,p]\quad X[1,p,1]\quad G[1,p,1]\ldots X[1,p,q]\quad G[1,p,q]$

$Y^{1}[2]\quad \theta[2,1]\quad X[2,1,1]\quad G[2,1,1]\ldots X[2,1,r]\quad G[2,1,r]$

.
.
.

$Y^{1}[t]\quad \theta[t,p]\quad X[t,p,1]\quad G[t,p,1]\ldots X[t,p,s]\quad G[t,p,s]$

$Y\ [t+1]$ (both flags set)

where

G,X,Y - Two bytes each, right justified.

$0 \le G \le 255$, optionally present

$0 \le X \le 4095$

$0 \le Y \le 4095$

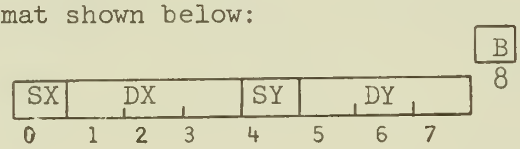
θ - Two bytes, right justified in the left byte.

$0 \le \theta \le 255$, optionally present

The beginning of each line has the flag bit set on the first byte of the first coordinate; following the last full line is a single coordinate with both flags set. When reading from the scanner in coordinate mode, flags can occur on the second byte of the position coordinate (X[i,j,k] in the above example format) if a buffer overflow occurs in the S-M-V Controller. Such a flag indicates that one or more additional hits occurred while this coordinate information was still in the S-M-V Controller buffer; hence, these additional hits were not recorded.

3.6.2.2 Incremental Format

The incremental string is composed of a sequence of bytes having the format shown below:



If (DX,DY) ≠ (0,0), the byte is interpreted as a segment vector consisting of two incremental displacements, DX and DY, the corresponding signs for each displacement, SX and SY, and the beam condition, B.

DX,DY - three bits each, right justified.

$0 \leq DX \leq 7$

$0 \leq DY \leq 7$

SX,SY - one bit each.

0 = plus

1 = minus

B - one bit.

0 = beam off

1 = beam on

If (DX,DY) = (0,0), the two signs (SX,SY), and the beam condition, B, are interpreted as an incremental command with the format and meaning as follows:

SX	SY	B	Mnemonic	Definition
0	0	0	NOP	No OPeration
0	0	0	H	Halt, close out the operation
0	1	0	RGR	Reset Grid Resolution
0	1	1	RVB	Reset Vector Begin point
1	0	0	RSS	Reset Stencil Size and/or shape
1	0	1	RSO	Reset Stencil Orientation
1	1	0	MB	Modulate the Beam (at a fixed position)
1	1	1	IT	ITerate (magnify) the next segment vector

The four commands which reset parameter values are followed by the bytes containing the new parameter values: one byte each for RGR and RSS and four bytes each for RVB and RSO. The byte format for each of these is the same as that defined for the initializing parameter string.

The IT command is followed by two bytes: the first is the count and the second is a segment vector. The effect of this command is to repeat the given segment vector as though this segment vector had appeared sequentially in the input string the number of times given in the count byte. If the displacement of the segment vector is (0,0) or if the count is 0, the IT command is ignored.

3.6.2.3 Raster Format

The raster data string used by the S-M-V controller is similar to that used by the PAU. The format is as follows:

$$\begin{array}{llll}
G_{111} & G_{112} & \cdots & G_{11n}^{1} \\
G_{121} & G_{122} & \cdots & G_{12n}^{1} \\
\vdots & & & \\
G_{1p1} & G_{1p2} & \cdots & G_{1pn}^{1} \\
G_{211} & G_{212} & \cdots & G_{21n}^{1} \\
\vdots & & & \\
G_{tp1} & G_{tp2} & \cdots & G_{tpn}^{1}
\end{array}$$

The string is packed and consists of 1,2, 4 or 8 bit characters (G_{ijk}) depending upon the gray level encoding.

The flag bit of the last byte for each orientation sampling of a scan line is 1. If the last character, G_{ijn}, does not end on a byte boundary then the remaining bits in the byte are set to zero.

Parametric description of a scan-display system *

by LAWRENCE A. DUNN, LAKSHMI N. GOYAL,
BRUCE H. McCORMICK and VAL G. TARESKI

University of Illinois
Urbana, Illinois

Reprinted from –
AFIPS – Conference Proceedings, Volume 34
Copyright © by AFIPS Press
Montrale, New Jersey
07645

INTRODUCTION

Automatic pattern recognition and graphical data pro-essing have recently received considerable attention. n addition to analysis and processing of pictorial in-ormation, there is a need for interactive display systems o present both intermediate and final processed data. The subject of graphic display terminals has been xtensively discussed in the literature. Scan-display ystems oriented towards image processing, however, vith particular attention to bypassing the central rocessing unit for as many tasks as possible have not ad comparable development. This paper is focused on his latter area.

A computer system for image analysis and display has hree principal constituents: image acquisition and isplay, image encoding for digital transmission, and nally procedures for classification of the encoded nage. A new direction in image acquisition and display s to append a *Video Communications Net* to the central omputer so as to provide the remote users with video ransmission to centralized image processing facilities.

Requirements on image encoding for transmission f information are dependent upon application.[2] For xample the bubble chamber data processing of high nergy physics requires a very high positional resolu-on, although little demand is made on the gray-scale ontent of the picture. In the environmental sciences, owever, gray-scale resolution is critical. Biomedical pplications normally place stringent requirements on ray-scale resolution, but high positional resolution of the image is not required. Several systems are operative in high energy physics, such as PEPR[3] at M.I.T., CHLOE[4] and POLLY at Argonne National Laboratory, HPD[5] at Brookhaven National Laboratory, and HUMMINGBIRD[6] at Stanford Linear Accelerator Center. Other systems are FIDAC[7] in biomedicine and KARLSRUHE[8] in automatic photointerpretation. However, we feel that these systems are unnecessarily specialized, and there is need for a more versatile and general system applicable to diverse disciplines on an integrated basis and amenable to modification as the need arises. In this paper we define an integrated system for the image acquisition and display.

Figure 1 shows the proposed system. Video switching matrix provides the facilities for remote users. High resolution CCTV cameras are provided for image encoding and acquisition. Remote video consoles, consisting of two high resolution monitors and a teletype set, provide information display at the remote user's end. Videograph printer outputs a facsimile copy at video rates, where the copy can have any admixture of text, graph or half-tone pictures. Microimage store provides the system with an extensive store of images—as direct images and not as digital data—and finds application in information retrieval areas such as library automation and biomedicine, where there is need for a permanent huge mass storage. High resolution scanners allow the scanning of the film for accurate measurement purposes and also allow the construction of images on film. High resolution monitors are slaved to the scanner system in a manner which allows the monitors to borrow inexpensively the scanner control. If video scan converters are part of the video switching matrix, video images can be treated as if they were on film and thus the same encoding techniques and the same programs could be exploited for both without

Supported by Contract AT(11-1)-1018 with the U.S. Atomic nergy Commission and by the Advanced Research Projects gency.

ased on an internal document published in the Department of omputer Science.[1]

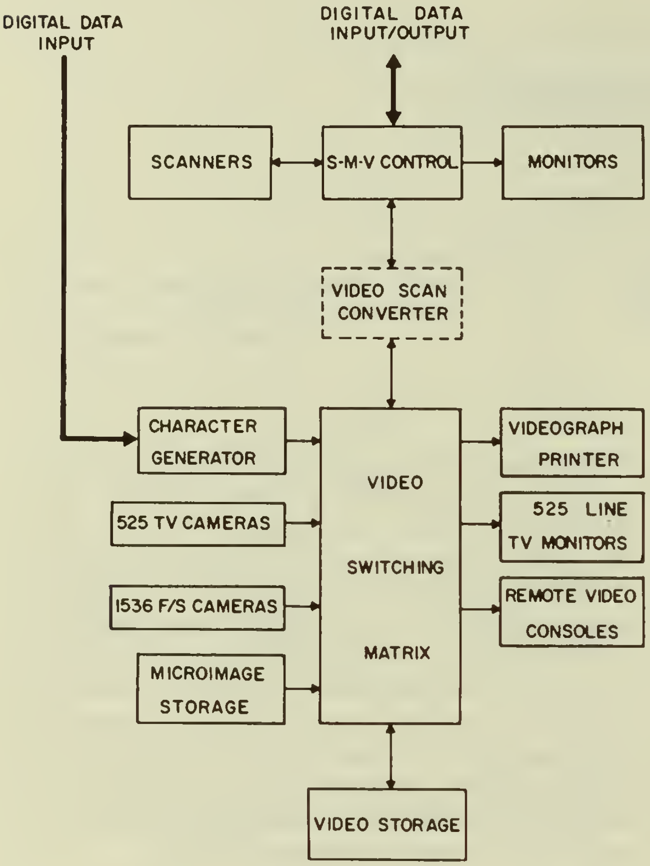


Figure 1—Block diagram of the scan/display system. Note that the video scan converter can be bypassed under program control

program change. However, the video scan converter is not essential, since the controller can handle constraints of the video system with some sacrifice in resolution. Character generator provides facilities for message handling and display in general, particularly for CAI, which along with the scanner provides the flexibility of displaying line drawings and half tone pictures intermixed with the text. More details about the devices shown in Figure 1 are given in Appendix A.

Emphasis in the paper is given to inter-media transformation options—translation, magnification and rotation—that can be effected by control of position counters, to constraints necessary for maintaining media compatibility, and to alternate digital representations of an image. The major goal of the paper is to abstract the system parameters and to develop the relations among them. These system parameters are listed with their definitions in Appendix B.

Design values of these parameters for the Illiac III Scan-Display System are tabulated in Appendix C. For this system the format of the scan/display parameters is shown in Figures 18 and 19.

Ways of digitally representing an image

By selecting a grid and associated coordinate system an image can be represented as a digital string of encoded local samples. The coordinate system allows specification of a sampling position, or by implication a sampling sequence; and the grid mesh allows specification of a sampling resolution, or by implication a sampling frequency. One conventional image digitization procedure specifies sequential sampling along successive lines parallel to a specified coordinate axis. To describe these linear sweeps we will use the terms 'scan line' and 'scan axis.' Although several string definitions are possible each will always contain, either explicitly or implicitly, two correlated types of information:

the coordinates of a point, and

the image density level in some localized area about the point.

If the localized sampling area is not circular in shape then a third type of information can be included

local orientation of the sampling area.

The encoding/decoding format for these three items (position, density and orientation) is invariant in any given string definition. Any further interpretation of the string requires knowledge of the sampling technique employed.

Specifying a sampling sequence, a digital string representation can be formed in two different ways

(r) encoding the image density level for each coordinate in the sequence, or

(c) forming an ordered substring of coordinates by selecting only those from the given sequence for which the image density level satisfies some prescribed criteria.

For a particular image each of the two strings is unique within the reproducibility limits of density detection and encoding.

String (r) is usually more representative and economical (in terms of string length) for images with a frequently varying density, i.e., with large amounts of detailed information—as a page of text with illustrations or a stained tissue section.

String (c) is more economical and representative for sparse images such as an engineering drawing or bubble chamber negative. For these images high resolution is needed along the scan axis but the scan lines can be relatively far apart without loss of information. A rectangular or slit-like sampling area is effectively employed if its orientation is within approximately 45° of being perpendicular to the scan line. However, to sample all possible orientations it is necessary to:

(c1) sample each scan line several times—once for each of a sequence of orientations,

(c2) sample a second sequence in which scan lines are perpendicular to those of the first.

String (c) can be a function of the sampling technique, i.e., one can sample *for* and recognize *only* a preselected class of features. Two possible classes are:

(c3) silhouettes, or images containing large areas of uniform density—adjacent areas are distinguished by a sharp discontinuity in image density. Chromosome karyotyping falls in this class.

(c4) outlines, or sparse line drawings—these are a special case of silhouettes. The engineering drawings and bubble chamber negatives are in this class.

Sampling criteria corresponding to each of these classes can be stated as:

(c3a) select the point if the *change* in image density between it and the previous point(s) exceeds some threshold.

(c4a) select the point if the image density *increases above and then decreases below* some threshold between it and the previous point(s).

The image density level in each case can be recorded in the string representation along with the coordinates. If it is known that the image contains line (or boundary) information—perhaps from having processed a string obtained in one of the aforementioned ways—then it may be useful to track the constituent lines by sampling a sequence of localized areas with restricted sets of orientations. This tracking procedure then generates a sequence of head-to-tail vectors or, in the limiting case, a string of incremental displacements. This concept is also useful in creating an image by first constructing the incremental string. A natural interpretation of the string is:

(i) an ordered set of commands defining starting point, line width, line intensity and the (incremental) segment vectors.

Curved lines obviously can be represented as a sequence of sufficiently short segmental displacements.

These three ways of forming strings—(r), (c) and (i) are correspondingly defined as the *Raster*, *Coordinate*, and *Incremental* data formats.

Raster format

Stated in PL/1 the sampling algorithm for an X-axis scan including the optional orientation sampling is:

```
    IF¬SLIT then θB = θE;
SY: DO   Y = YB BY ΔY TO YE;
Sθ: DO   θ = θB BY Δθ TO θE;
SX: DO   X = XB BY ΔX TO XE;
        'encode/decode G(Y, θ, X)';
        END SX;
        END Sθ;
        END SY;
```

where (XB, YB) and (XE, YE) define a rectangular area to be sampled at increments of (ΔX, ΔY). Each scan line is sampled once for each orientation from θB to θE at increments of Δθ.

The total number of samples is:

N_s = Number of lines × number of orientations per line × number of samples per orientation

and the number of bits required for storage is:

N_b = Number of bits per sample × number of samples.

Coordinate format

Stated in PL/1 the sampling algorithm for an X-axis scan is:

ENCODING (READ)

```
    IF¬SLIT THEN θB = θE;
SY: DO   Y = YB BY ΔY TO YE;
Sθ: DO   θ = θB BY Δθ TO θE;
SX: DO   X = XB BY ΔX TO XE;
    IF 'Criteria satisfied' THEN
CYES: DO;
        IF 'First time for this Y and θ' THEN
        FYES: DO; 'output Y';
```

```
            IF SLIT THEN 'output Θ';
            END FYES;
        FNO: DO; 'output X';
            IF NGL >2 THEN 'output G';
            END FNO;
        END CYES;
    CNO: END SX;
        END SΘ;
        END SY;
```

DECODING (*WRITE*)

```
INIT: 'input YIN';
      IF SLIT THEN 'input ΘIN';
      'input XIN';
      IF NGL >2 THEN 'input G';
SY:   DO   Y = YB BY ΔY TO YE;
SETΘ:Θ = ΘIN;
      IF Y = YIN THEN
      SX: DO X = XB BY ΔX TO XE;
          IF X = XIN THEN
      MOD: DO; 'modulate beam';
          'input next coordinate';
          IF 'coordinate is a new Y' THEN
      NEWY:DO; IF SLIT THEN 'input ΘIN';
          'input XIN';
          IF NGL >2 THEN 'input G';
          GO TO SETΘ;
          END NEWY;
      NOTNEWY: IF NGL >2 THEN 'input G';
          END MOD;
          END SX;
          END SY;
```

where (XB, YB) and (XE, YE) define a rectangular area to be scanned at increments of (ΔX, ΔY). In encoding each scan line is swept once for each of the orientations between ΘB and ΘE, where ΔΘ is the orientation increment. Actual encoding takes place only when the criteria is satisfied. In decoding, only those scan lines which are specifically read in are swept at the input orientation, and writing takes place (or more generally, is initiated) only at those positions which match the input coordinates.

NGL is the number of gray levels. If this number is greater than two, additional encoding/decoding is necessary to obtain the gray levels. For NGL = 2, the fact that the criteria is satisfied implies the gray scale information.

The most general coordinate string of an X-axis scan has the form:

Y coordinate, Θ, X coordinate, gray scale, X coordinate, gray scale. . .

Y coordinate, Θ, X coordinate, gray scale, . . .

Increment format

The incremental string is composed of a sequence of elements that can be interpreted either as a segment vector or as an incremental command. The segment vector is composed of two incremental displacements, DX and DY, the corresponding signs for each displacement, SX and SY, and the 'beam condition.' DX specifies the number of unit cells the beam is to be displaced in the X direction and DY specifies the number of unit cells the beam is to be displaced in the Y direction. The beam condition is given as being either on or off during the move.

If (DX, DY) = (0, 0), the two signs, (SX, SY), and the 'beam condition' are interpreted as an incremental command, where incremental commands have the following semantics:

H *H*alt, close out the operation.
IT *IT*erate (magnify) the next segment vector. The next two elements in the string should be a count followed by a segment vector.
MB *M*odulate the *B*eam intensity (at a fixed position).
NOP *N*o *OP*eration
RGR *R*eset *G*rid *R*esolution
RSO *R*eset *S*tencil *O*rientation
RSS *R*eset *S*tencil *S*ize and/or shape
RVB *R*eset *V*ector *B*egin point

The IT command with its count is equivalent to having the same segment vector appear sequentially in the string by the number of times given in the count byte. If the displacement is (0, 0) the IT command is ignored.

The four commands that reset parameter values are followed by string elements containing the new parameter values. The element format is the same as that defined for the initializing parameter string.

For the incremental format (XD, YD) and (XE, YE) are interpreted as defining a file area, outside of which no recording (film) or displaying will be allowed to take place. (XB, YB) becomes the initial point from which the beam will start the first segment vector.

Figure 2 shows the use of incremental vectors with the command RSS inserted at point P (between vectors (2,2) and (3,1)) and with command RGR inserted at point Q.

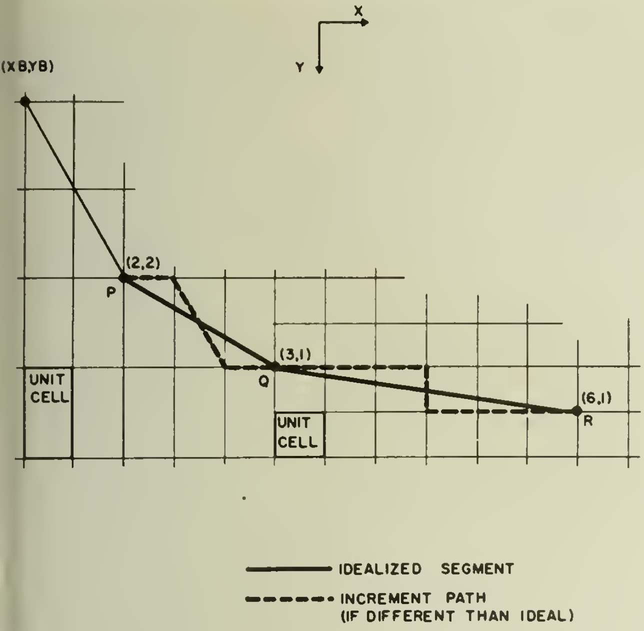


Figure 2—Segment vector plotting. Note change in stencil size at P and change in resolution (unit cell) at Q

Resolution, sampling and scanning parameters

Resolution

In order to encode a digital representation of an image t is necessary to impose a grid and coordinate system ıpon it. It is convenient to conceive the *total image or aster area as a unit square* and to interpret the address-ıble positions of the image as fractional coordinates anging between 0 and 1. The mesh of the grid uperimposed on this range is then 2^{b_g} where b_g is the ıumber of bits used to specify coordinate position. The smallest resolvable square has sides of 2^{-b_g} *units* nd is termed a *gross basic cell.*

The aspect ratio of the raster area need not be 1 : . The physical interpretation of the basic cell will ıore generally be a rectangle, hence the effective reso-ıtion along one axis may differ from that along the ther.

The adopted coordinate system—left-handed rec-angular—is a natural one for most textual material nd also corresponds to standard video practice of ft-to-right top-to-bottom scanning.

In concept one achieves the physical limit of resolu-on by choosing b_g sufficiently large. In practice this difficult to implement and one distinguishes between gross position counter specified by b_g, and a vernier osition counter specified by b_v. The vernier counter as a sign bit, s_v, and is interpreted as a signed position lative to the gross position. This is equivalent to verlaying a vernier grid in the immediate vicinity of a gross coordinate position (which can be interpreted as a local 'benchmark'). One then has a localized grid of mesh $2^{b_g+b_r}$ where b_r is the number of bits in the vernier counter used to extend the gross resolution. The smallest resolvable square has sides of $2^{-(b_g+b_r)}$ *units* and is termed a *vernier basic cell.* The remaining b_o bits in the vernier counter define the gross-vernier overlap, or the *maximum vernier window* as having sides of 2^{b_o} *gross basic cells* (see Figure 3) or $2^{(b_o-b_g)}$ units.

The above discussion defines the design parameters that determine maximum position resolution. Not all applications warrant this maximum resolution. More importantly one cannot contrive efficient scanning-recognition algorithms without a range of resolution options. One clearly wants independent choices of resolution for the two axes, either because the application warrants it, or becuase the format warrants it, as in the case when scanning in the coordinate format using a slit-like sampling area.

The parameters p and q specify the sampling resolution as every 2^p basic cells in the X-direction and every 2^q basic cells in the Y-direction:

$$\Delta X = 2^p \text{ basic cells} = 2^{p(b_g+k)} \text{ units} \tag{1}$$

$$\Delta Y = 2^q \text{ basic cells} = 2^{q(b_g+k)} \text{ units} \tag{2}$$

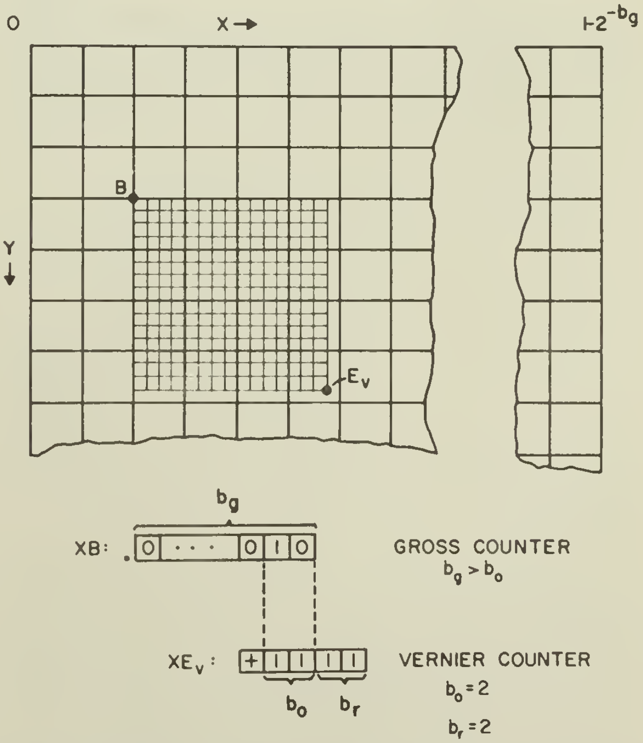


Figure 3—Maximum vernier window area with respect to B coordinate. Shown is an overlap of two bits (b_o) and maximum local resolution for a resolution extension of two bits (b_r)

The gross/vernier selection determines k as 0/b_r. The smallest resolvable rectangle is ΔX *by* ΔY units and is termed the *unit cell.*

The position counter is incremented at the p^th or q^th significant position of the gross/vernier counter, hence the number of significant position bits is:

	X	Y
raster gross:	$b_g - p$	$b_g - q$
raster vernier:	$b_g + b_r - p$	$b_g + b_r - q$

For the coordinate representation it is desirable to increase gross sampling resolution along the scan axis while incrementing the position counter at the p^th significant position as described above. This can be done by interpolating between counter increments and concatenating the b_c interpolation bits to the $b_g - p$ significant counter bits. One then has for the coordinate resolution:

coordinate gross: $b_g + b_c - p$ significant bits.

A natural choice is to make $b_c = b_r$ since b_r reflects the physical limit of resolution.

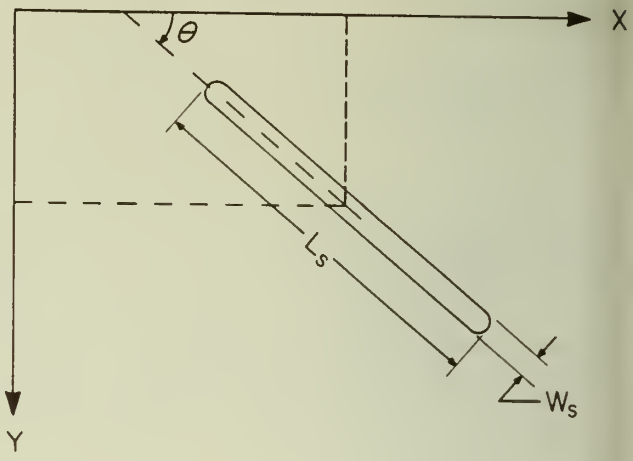
Sample encoding

The maximum number of image density states for a read command or recording beam intensity states for a write command are indicated by the gray scale parameter, n. The number of encoded bits is interpreted as 2^n, hence the number of possible states as 2^{2^n}. The maximum value of n is specified as n_{max}.

Triggering or filtering of output information may be done by either a standard level discriminator or by a specially designed plug-in unit. The value assigned to the parameter T chooses between these two.

The parameter B_s will distinguish between the choice of a standard size spot ($\simeq$1 gross basic cell in diameter), and a slit or non-standard spot size. If the non-standard option is taken, then the parameters u and v define the sample width and length as 4^u and 2^v gross basic cells respectively. The range of circular spot sizes can be specified by u with v = 0.

If the sample area is slit-like then the orientation becomes significant. The *unit of angle* is the circle, or radians/2π. The angular resolution is 2^{-b_a} units where b_a is the number of bits used to specify an angle. The angular sweep Begin-End coordinates, (θB, θE), specify the range over which incrementing is to be accomplished. The increment is defined by the parameter z as $\theta = 2^{z-b_a}$ units. The slit is swept the length of a scan line for each value of θ in the specified range.



SLIT:	SPOT:
$W_s < L_s$, θ RELEVANT	L_s = MINIMUM W_s = DIAMETER θ IRRELEVANT

Figure 4—Slit/Spot geometry

Figure 4 illustrates the geometrical definition and angular reference.

Scanning rate

Sweep velocity will in general be limited by the following:

1. positional digital-to-analog response time
2. maximum channel data transfer rate
3. number of bits of gray scale encoding/decoding

Sweep velocity can generally be expressed as a function of the following parameters (see Appendix B), where a constant clock rate for incrementing the positional counter is assumed:

$$V_s = c_1 \cdot f(p,q,A) \cdot g(DF, K, n, n_{max}).$$

Here f is determined by the scan axis and unit cell selection:

$$f(p,q,A) = (1\text{–}A) \cdot 2^p \cdot \delta x + A \cdot 2^q \cdot \delta y.$$

The maximum *continuous* data rate is required for the Raster format with n = n_{max}, where g can be normalized as

$$g = 2^{-n_{max}}.$$

The data per sample are higher for the coordinate format, but the average data rate is normally less than for Raster. (Otherwise the image would be more optimally encoded in a raster format.) Buffering is needed to handle local bursts of perhaps three or four consecutive samples. The amount of data for coordinate representation are essentially (though not absolutely) independent of the number of bits of gray scale encoding/decoding.

For an X-axis scan at *constant sample rate* one then has

$$V_s = c_1 \cdot 2^{-n_{\max}} \cdot 2^{p} \cdot \delta x.$$

c_1 can then be determined from the clock frequency. This equation yields a constant sample rate with the Raster format for any value of n.

Since the data per sample is 2^n bits, one obtains a *constant data rate* by introducing the quantity $n_{\max} - n$ in the exponent yielding:

$$V_s = c_1 \cdot 2^{-n_{\max}} \cdot 2^{p+(n_{\max}-n)} \cdot \delta x.$$

Here instead of incrementing the position counter at the pth significant position, the burden on the positional D/A conversion is appreciably lessened if one increments only at the $[p + (n_{\max} - n)]^{th}$ significant position and interpolates to obtain samples at the $(2^{n_{\max}-n} - 1)$ positions between counter increments. The constant data/sample rate option for a given resolution is then determined by the parameter K.

Window area

The term *window* was introduced above in defining the maximum area that can be covered in a single operation with vernier resolution. Two pairs of coordinates serve to specify the portion of an image to be scanned, encoded and displayed. They are termed the Begin coordinates (XB, YB), and End coordinates (XE, YE), and are interpreted as defining diagonally opposed corners of a rectangle. Since the position counters may be decremented as well as incremented any one of four B-E combinations may be chosen to specify the same window area. This feature and the coupled-uncoupled option of the monitor position counters allow the Rotation Group transformations described below. The coordinates must be multiples of (ΔX, ΔY).

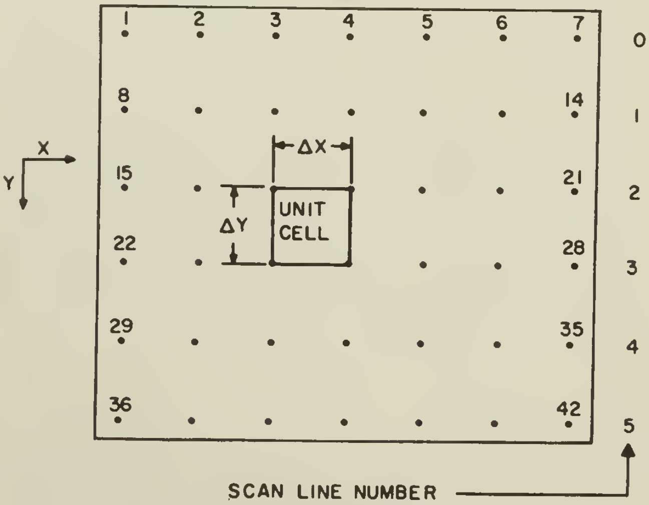
Scan format (lattice and sequence)

The scan format, determined by the parameters *L*, *S* and *A*, imposes a *Lattice* upon the grid in terms of the *unit cell* and specifies the scan line sampling sequence. Tho scan axis is specified by A as X or Y, and S selects the scan line sequence as interlaced or sequential. These two parameters along with the B-E combination completely determine the sampling sequence. One obviously starts with the Begin coordinate and terminates with the End coordinate, determining the direction of sampling along a scan line. The hexagonal/rectangular lattice option is defined by L as described below.

The matrices in Figures 5 and 6 illustrate the sampling sequences for a scan direction parallel to the X-axis on a *Rectangular Lattice.* The position of point No. 1 in the lattices is specified by the *Begin Coordinates.* The *End Coordinates* specify point No. 42 in the *Sequential* case (Figure 5) and point No. 15 in the *Interlaced* (Figure 6) case. Since the Begin and End coordinates are constrained to be multiples of ΔX and ΔY they will always be member points of the Lattice.

Figure 7 illustrates the relative positioning of points and their sampling sequence for the two scan directions on *Hexagonal Lattices* when line sampling is *sequential.* The Hexagonal Lattice is obtained by shifting the lattice points of *odd* numbered scan lines in the corresponding *Rectangular Format* by an amount ΔX/2 (or ΔY/2) along the positive scan axis. Otherwise hexagonal formats are analogous to their rectangular counterparts.

The *Interlace* option and the adopted coordinate system are particularly suited for communication with a standard video network through an analog-to-digital interface. Figure 6 shows the relation between the



SEQUENTIAL: LINE SEQUENCE IS 0,1,2,3,4,5

Figure 5—Sequential scan along X-axis using a rectangular lattice. (ΔY = ΔX)

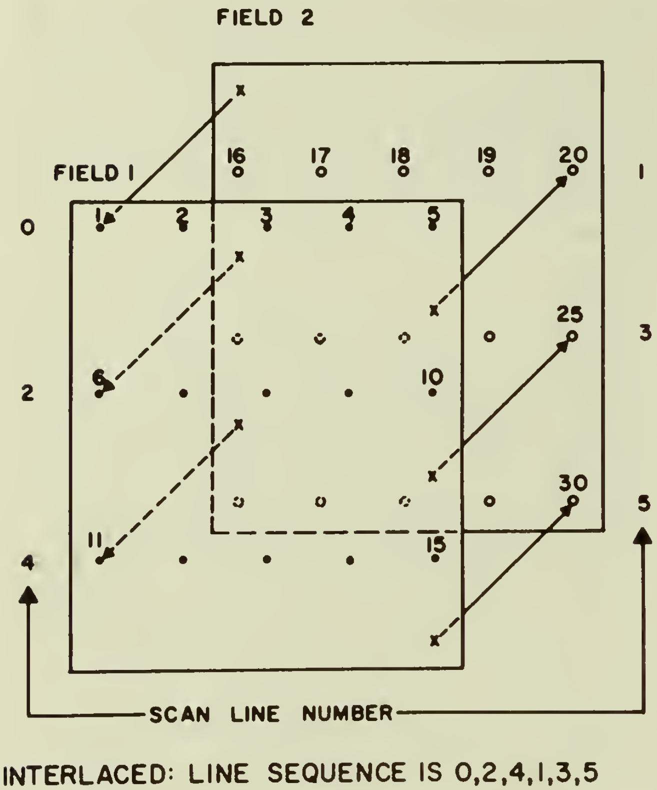
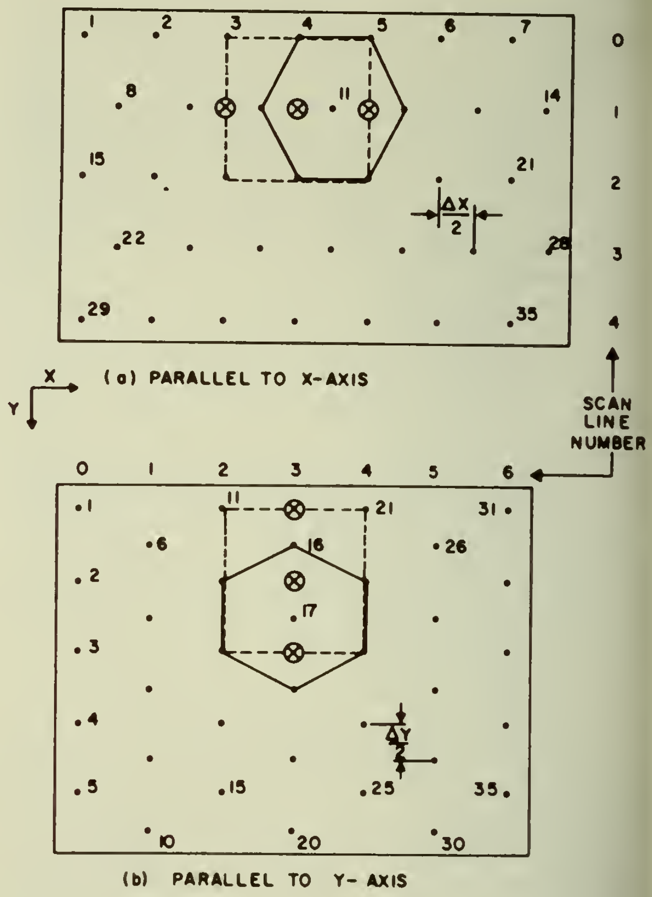


Figure 6—Interlaced scan along X-axis using a rectangular lattice. (ΔY = ΔX)



(a) PARALLEL TO X-AXIS

(b) PARALLEL TO Y-AXIS

Figure 7—Sequential scan along (a) X-axis, and (b) Y-axis using a hexagonal lattice. The hexagons around point 11 in (a) and point 17 in (b) illustrate neighbor points. The dotted rectangles show neighbor points in the corresponding rectangular lattice. (ΔY = ΔX)

fields and the sample points for the interlace option.

Lines have been drawn between points surrounding point No. 11 in Figures 7(a) and 8 and point No. 17 in Figure 7(b) to illustrate the concept of neighbor points. An interior point of the Hexagonal Lattice has *6* neighbor points whereas that of the Rectangular has 8. Since the Hexagonal Lattice is not regular (it is rhombic), although it is nearly so for ΔX = ΔY (see Figure 7), neighbor points are *not* all equidistant from their interior point; but they always partition by distance into two sets of 4 and 2 points each. Those for the Rectangular Lattice partition generally into three sets of 4, 2 and 2 points each, and for ΔX = ΔY the two sets of 2 and 2 become a single set of 4. Compare Figure 8 with Figure 7.

Video compatibility

When interfaced by a video scan converter the video network can be handled as a scanner. However, the scan converter is not essential since the S-M-V Controller can be designed to satisfy the constraints of the video systems. The following discussion uses the parameter i to identify the video system within the network;

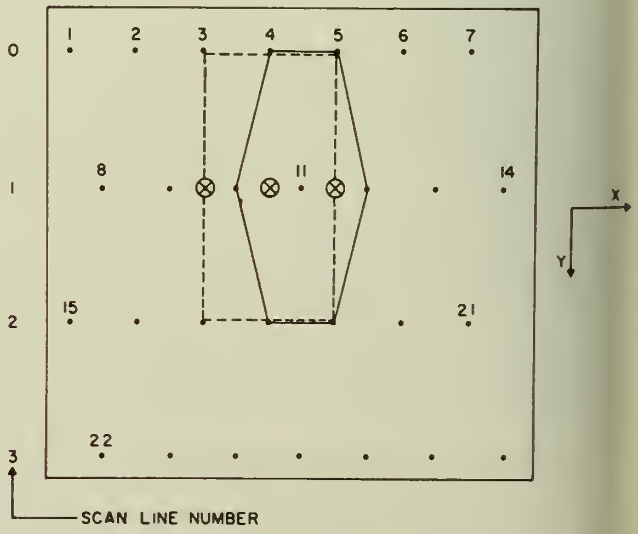


Figure 8—Same as Figure 7 (a) with ΔY = 2ΔX

it assumes the controller-video link to be direct and develops the corresponding constraints.

Since the viedo line sweep timc, $\Delta t(i)$, is a fixed parameter the number of data bits that can be transferred to or from core per full scan line is constrained by the I/O channel capacity, where the maximum bit transfer rate is f_b. A buffer will allow a 'burst-mode' sampling for some fractional part of the scan line—hence higher resolution in sampling a vertical band can be achieved.

The Incremental string cannot be passed diiectly to video; it must first be passed to a scan converter.

The Coordinate string with video has a resolution along the scan line equivalent to Raster resolution with $n = n_{max}$, hence the scan axis counter increment, ΔC, is given by:

Rastei with constant data rate option:

$$\Delta C = 2^{p+(n_{max}-n)}$$

Raster with constant sample rate and Coordinate:

$$\Delta C = 2^p.$$

One would normally choose the constant data rate option. The Coordinate string resolution can only be locally maintained because the buffer stoiage limits the number of successive samples in contiguous unit cells. This is not a severe restriction since the coordinate representation is not very meaningful unless the image is rather sparse. Note that orientation sampling is meaningless with video.

The entire video discussion is from the point of view of the Raster string representation. The stiing will have existence only when core memory participates; otherwise there is only the analog signal transfer between the other three media. The resolution results are valid independently of the point of view.

Horizontal (X-axis) resolution

To achieve a maximal uniform resolution across a full scan line it is necessary to maintain a constant data rate. We choose the largest j such that

$$2^j \le f_b \,.\, \Delta t(i) \tag{3}$$

and hence maximize and fix the number of bits in a full line of sampling. Position resolution and gray scale resolution are not independent: the number of bits per sample is 2^n, hence equation (1) constrains the number of samples in a full line to be

$$S(j,n) = 2^{j-n}\,. \tag{4}$$

Maximizing position resolution minimizes gray scale resolution and vice versa. S(j,n) achieves its maximum value for $n = 0$:

$$S_{max}\,(j) = S(j,0) = 2^j \tag{5}$$

Table I shows values of $S_{max}(j)$ for three different video systems. Parameter values used in the calculation are given in Appendix C. If a window contains only a fractional part of a scan line, then only a corresponding fractional part of S(j,n) samples can be obtained along each scan line.

Sampling at the position counter stepping frequency, f_c, maintains the aforementioned data rate for an $2^{n_{max}}$ bit gray scale datum. An interpolation counter is used to achieve the rate for other choices of image density resolution with the constant data rate option:

$$\Delta C = 2^{p+(n_{max}-n)}.$$

A 1-1 correspondence between a full scan line at video and the S-M-V control grid requires:

$$S(j,n) \cdot \Delta X = 2^{b_0} \text{ basic cells.}$$

Substituting for S(j,n) from equation (4) yields:

$$(\Delta X)_{max}\,(j,n) = 2^{b_0-(j-n)}, \text{ hence}$$

$$p_{max}(j,n) = b_0 - (j\text{-}n). \tag{6}$$

This defines the achievable video resolution along

Table I—Inherent characteristics of three video systems and maximum sampling resolution, $S_{max}(j)$, as constrained by other media

i	$N(i)$	$\Delta t(i)$ μ-sec	$S_{max}(j)$	j
1	525	≈ 56	512	9
2	1536S	≈520	4096	12
3	1536F	≈ 43	256	8

F = FAST SCAN
S = SLOW SCAN

Table II—Maximum sampling resolution and interdependence of position and gray scale resolutions as constrained by media characteristics. The main entry is the maximum number of samples per video scan line, S(j, n), and the value in parentheses is the corresponding maximum value of ΔX. $(\Delta X)_{max}(j,n)$

j	n				g(j)
	0	1	2	3	
9	512 (8)	256 (16)	128 (32)	64 (64)	0
12	4096 (1)	2048. (2)	1024 (4)	512 (8)	-1
8	256 (16)	128 (32)	64 (64)	32 (128)	3
	1	2	4	8	
	2^n, bits of gray scale				

scan lines as well as the largest useable sampling increment that may be associated with the S-M-V control grid scan axis. Table II illustrates the interdependence of position and gray scale resolution for the three video systems of Table I. It contains the values of S(j,n) as constrained by equation (4), and in parentheses the corresponding maximum values of ΔX as given by equation (6). All parameter values used are listed in Appendix C. The function g(j) is explained in the following section.

Vertical (Y-axis) resolution

Having determined the video resolution along a scan line we now deterpine the vertical sampling so as to achieve a unit cell match to the S-M-V control grid. Assuming an X-axis scan at the controller, this constraint requires the video line sampling frequency to be

$$(\Delta Y)_v = \frac{r_v}{r_s} \cdot \frac{N(i)}{S(j,n)} \cdot \frac{\Delta Y}{\Delta X}, \qquad (7)$$

where r_s and r_v are the aspect ratios at the control grid and video, respectively. N(i) is the number of lines per video frame. The unit cell resolution ratio at the S-M-V control grid is $\Delta X/\Delta Y$, and the corresponding video resolution ratio is $1/S(j,n)\ N(i)/(\Delta Y)_v$. Using equations (1), (2) and (4) equation (7) can be written as

$$(\Delta Y)_v = r_v/r_s \cdot N(i) \cdot 2^{q+n-p-j},$$

which may be restated as:

$$(\Delta Y)_v = f(j) \cdot 2^{q+n-p},$$

where

$$f(j) = r_v/r_s \cdot N(i) \cdot 2^{-j}.$$

As in the discussion of horizontal resolution we wish to approximate by powers of 2, and determine a g(j) such that

$$f(j) \cong 2^{g(j)} .$$

The video line sampling frequency is then determined at the S-M-V controller from

$$(\Delta Y)_v = 2^{q+n+g(j)-p}$$

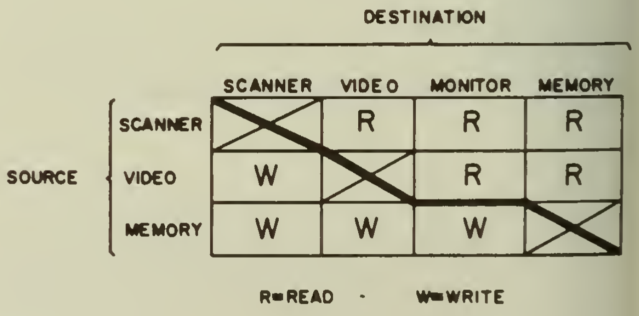
since the four terms in the exponent are defined by the parameter assignments.

Media selection

Media selection is determined by the parameters F, V and C. The choice of a Read or Write command is then determined from the source-destination matrix shown in Figure 9. Of the sixteen possible states for F, V, C and Read/Write ten are allowed as meaningful or useful, and this may be succinctly stated as:

States	*Source*		*Destination*		*Command*	*Figure*
4	(F⊕V)	.	(C⊕C)	.	READ	15
1	F	.	V.C	.	READ	16
1	V	.	F.C	.	WRITE	16
4	C	.	(F⊕F) · (V⊕V)	.	WRITE	17

where ⊕ means exclusive OR. As indicated these states are illustrated in Figures 15-17. A destination medium always exists since the monitor participates in all



		DESTINATION			
		SCANNER	VIDEO	MONITOR	MEMORY
SOURCE	SCANNER		R	R	R
	VIDEO	W		R	R
	MEMORY	W	W	W	

R=READ · W=WRITE

Figure 9—Command matrix. Read/Write selection as a function of selected media and desired transfer direction

operations. Whenever core memory (C) is *not* the image source, display at the monitor or video can be indefinitely repeated by setting the Regeneration parameter, J.

When transferring an image between two media, it is necessary to consider:

a. the X/Y aspect ratios, and
b. the X/Y resolution ratios.

If either of these ratios differ, then a contraction quite independent of any magnification can take place along one of the axes. Both sources of image distortion may be averted by matching *aspect ratios of the unit cell* at source and destination media. Since several media may be involved it is useful to adopt the concept of an S-M-V control grid and coordinate system through which any inter-media transfer must pass, as illustrated in Figures 14 through 17. One then forces a match between each medium and the S-M-V control grid. All position and resolution specifications in the parameters can be interpreted as referring to the S-M-V control grid and coordinate system. They must be specified with a particular medium (or media) in mind, however, and must be compatible with its associated characteristics. Scanner and monitor are completely dominated by the S-M-V Controller, hence the unit cell match is easily accomplished. The same is true of core memory as a destination medium. As a source medium the core memory unit cell match is under program control, and it is therefore necessary to associate inviolate unit cell as well as other parameter information with any string representation. Except for initiating a video scan, the link between S-M-V Control and video is basically an information transfer link. The video match is effectively accomplished by selectively transferring information from video (say every other scan line) and by holding back information to video (say blanking every other scan line) by employing $(\Delta Y)_v$ as discussed in the section on Vertical Resolution.

Extending this "match-to-control" concept allows all the transformations (rotation, magnification and translation) discussed in the following sections as well as the dual interpretations of video as a source medium.

The entire transformation discussion is from the point of view of gross resolution. Vernier resolution differs basically in having an inherent magnification of 2^{b_v} to both Monitor and Video, and in having a limited window area.

Monitor transformations

Information is communicated to the monitor during each of the S-M-V operations. The area of interest in an image is specified by a 'window' at the S-M-V control grid delineated by the Begin and End Coordinates. Three types of transformations may be applied to the window as viewed at the monitor: rotation, magnification and translation.

Rotation

The Rotation parameter, G, allows the option of:

(R1) slaving the monitor to the S-M-V Controller grid in scan axis and direction or,

(R2) choosing the scan axis and direction at the Monitor to be parallel to the X-axis and incrementing irrespective of the Controller choices.

When option (R2) is taken then scanning the X-axis at the Controller grid obtains the transformations shown in Figure 10, whereas scanning the Y-axis yields the transformations shown in Figure 11. The choice of a B-E orientation fixes the scan direction and the initial scan line, hence selects one of the four transformations. The four transformations in Figure 10 are called the "four group" of rotational symmetries on the rectangle. The eight transformations in Figures 10 and 11 define the "Klein Rotation Group" of symmetries on the square.

Magnification

The window displayed at the monitor can be magnified by factors of 2 with the parameter *h* subject to the combined restrictions:

$$h + \begin{Bmatrix} p \\ q \end{Bmatrix} \leq \begin{Bmatrix} p_{max} \\ q_{max} \end{Bmatrix}$$

The underlying assumption is that the range of resolution options on p and q is identical at monitor and scanner, and that the unit cell at monitor is magnified by $m = 2^h$.

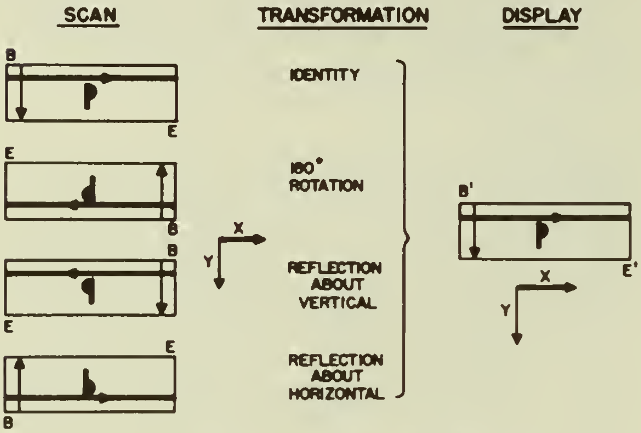
One naturally constrains the choice of *h* to keep the magnified window from exceeding the raster area:

$$m \cdot \left\{ \begin{vmatrix} XB\text{-}XE \\ YB\text{-}YE \end{vmatrix} \right\} \leq 1.$$

This restriction is refined in the discussion below on translation.

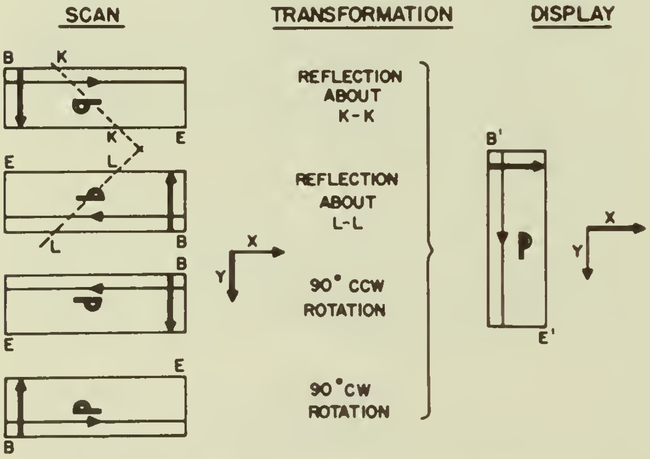
Translation

Translation of the window at the monitor may be achieved with the *Monitor Displacement Coordinates.*



DARK LINE AND ARROW SHOW SCAN AXIS AND DIRECTION

Figure 10—Group transformations effected by the four different Begin-End coordinate orientations with X-axis scan. (G = 1)



DARK LINE AND ARROW SHOW SCAN AXIS AND DIRECTION

Figure 11—Same as Figure 10 with Y-axis scan. (G = 1)

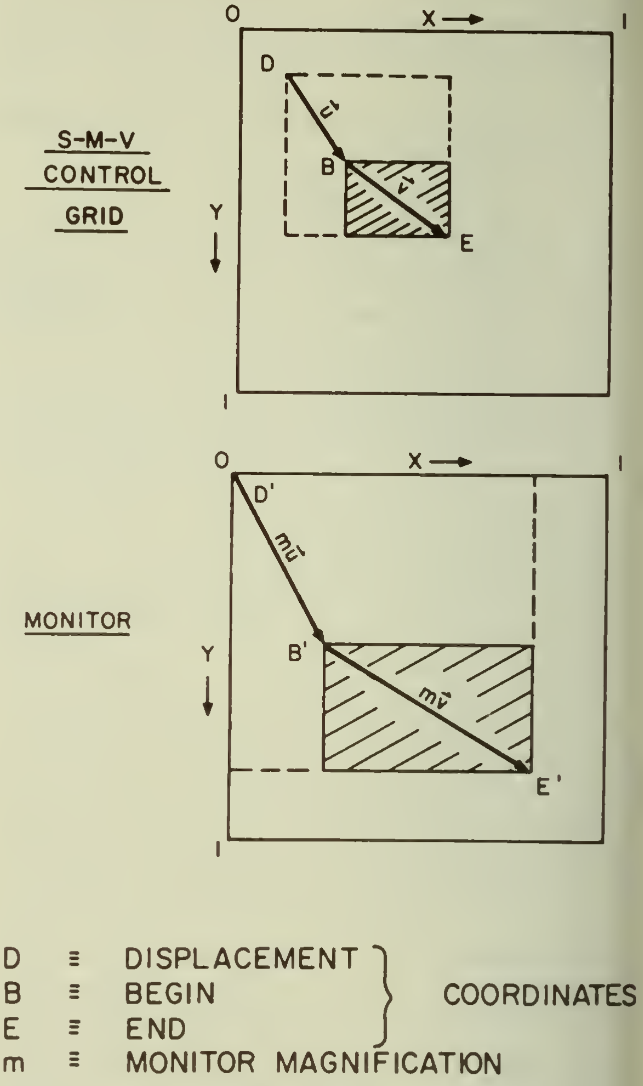


Figure 12—Image translation and magnification at the monitor

As shown in Figure 12, D transforms into (0, 0) at the monitor, hence B is repositioned accordingly.

If a magnification m is superimposed on the translation it affects the area delineated by the D-E coordinates, hence the combined transformation is completely defined as operating on the two vectors u and v:

1. translate the tail of u to (0,0) and
2. magnify the length of u and of v by m.

The four allowed orientations of D, B and E are shown in Figure 13. The implied constraint is that in case (a) D $\leq$ B $\leq$E with a corresponding interpretation for the other three cases. D always transforms into the corresponding corner position at the monitor with Rotation option (R1). For Rotation option (R2), D goes to (0, 0) at the monitor in all cases.

For proper centering, one must choose (XD, YD) such that

$$m\ (|XB - XE| + 2\,|XD - XB|\ ) = 1$$

and

$$m\ (|YB - YE| + 2\,|YD - YB|\ ) = 1.$$

Complete positioning freedom is not always possible when combined with one of the rotation transformations, e.g., when the window is very close to the edge of the grid and the chosen transformation requires D to be on the edge side of the window.

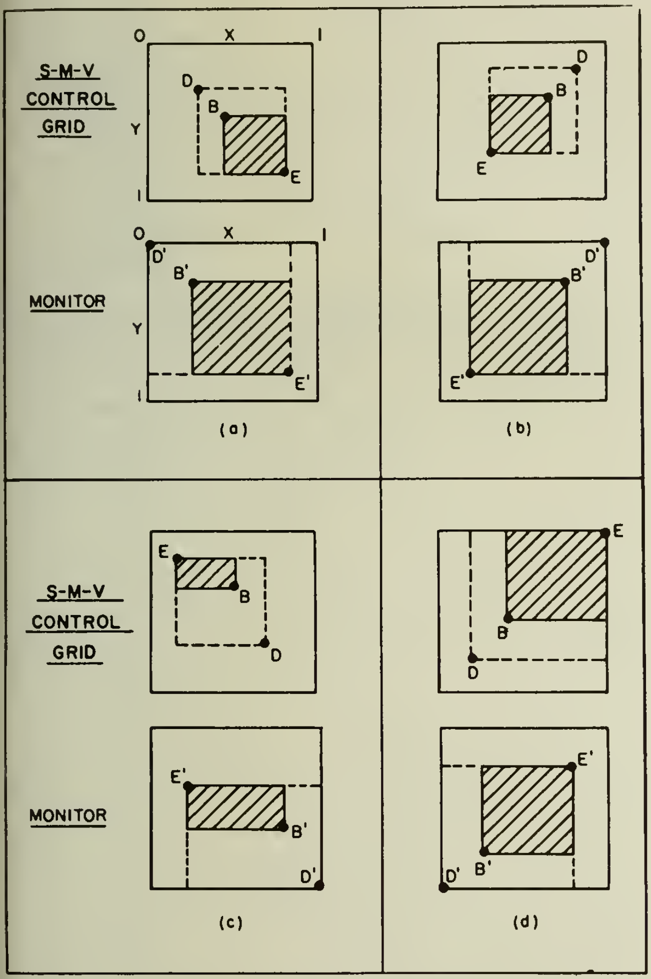


Figure 13—Four allowed orientations of D, B and E coordinates with the corresponding monitor interpretation. (G = 0)

Totally slaved to scanner

When tracking a line or a boundary, by scanning a sequence of windows, a 1-1 correspondence between Monitor and source is desirable; otherwise the relative positioning of windows at the source is not reflected at the Monitor, and the tracking procedure cannot be viewed. This can of course be achieved with the proper parameter assignments as a standard transformation but one would like to avoid the time involved in doing so. Since the window will be small the scanning time can be significantly reduced by recognizing a special case. The following natural setting for the parameters listed can be interpreted as defining the special case:

1. (XD, YD) = (0, 0), no translation
2. gross resolution
3. no rotation, option (R1)
4. no magnification

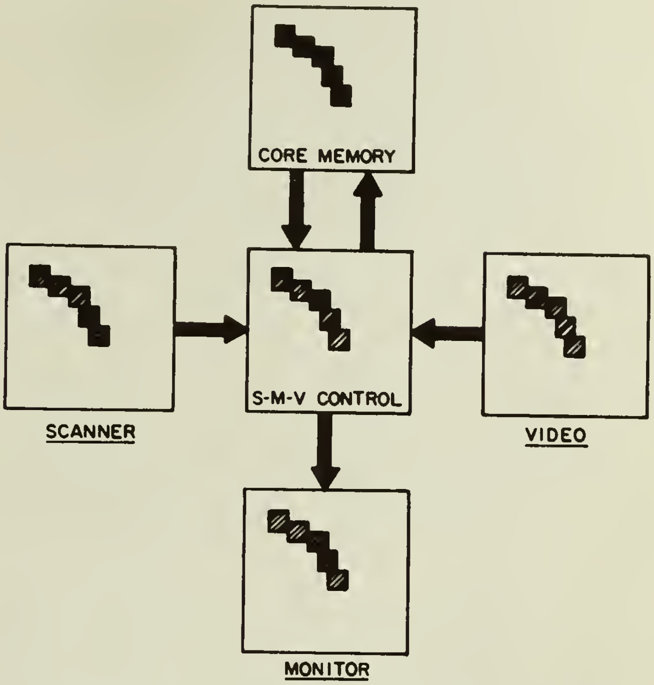


Figure 14—Monitor totally slaved to the source media

As shown in Figure 14, the Controller can then avoid the time-consuming redundancy of the transformation steps inherent in a sequence of small windows.

Totally slaved to video

As indicated in Figure 14, video is included in the total slaving concept. For video this is accomplished by replacing constraint (4) in the previous section with the following:

$$(4') \quad h \cdot \Delta X = \Delta X_{max}\,(j,n).$$

At most one window can be scanned per video frame, thereby limiting the repetition rate.

Video transformations

The video network, unlike the monitors, can act both as a source and as a destination. A window, specified by the Begin and End coordinates at the S-M-V control grid, determines the area of interest. Transformations similar to those at the monitor can be effected within the limits of the video constraints.

Source medium options

It is useful to distinguish two interpretations of video as a source medium:

(S1) 1-1correspondence between video and the S-M-V control grid (excluding rotation), thereby allowing translation and magnification of a window to the monitor;

(S2) 1-1 correspondence between video and some portion of the S-M-V control grid, effectively allowing translation and *de*magnification of a window to film.

Options (S1) and (S2) are illustrated in Figures 15 and 16 respectively.

Rotation

When video acts as a destination medium, the transformations are effected in the same manner as they are to the monitor under option (R2). Because the video scan axis and direction are fixed, option (R1) never applies.

When video participates as a source medium the role is reversed and one always gets the inverse transformation to the S-M-V control grid. If option (R2) is chosen for the monitor the two transformations cancel (into and then out from the control grid)—effectively yielding the identity transformation to the monitor.

Magnification and demagnification

Equation (6) defines $(\Delta X)_{max}(j,n)$, the largest useable ΔX. Choosing $\Delta X < (\Delta X)_{max}(j,n)$ allows a video magnification of

$$m_v(j,n) = \frac{(\Delta X)_{max}(j,n)}{\Delta X} = 2h_v(j,n) ,$$

so

$$h_v(j,n) = b_g - j + n - p = p_{max}(j,n) - p .$$

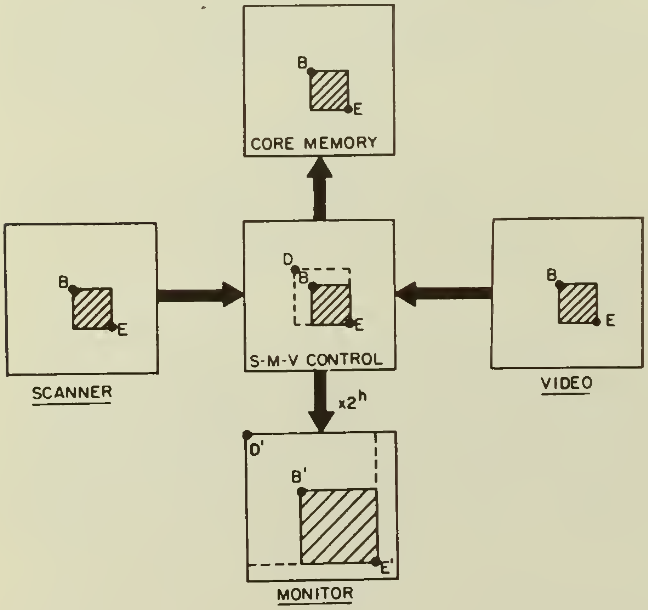


Figure 15—Magnifying a window at the monitor and/or transmitting to core memory

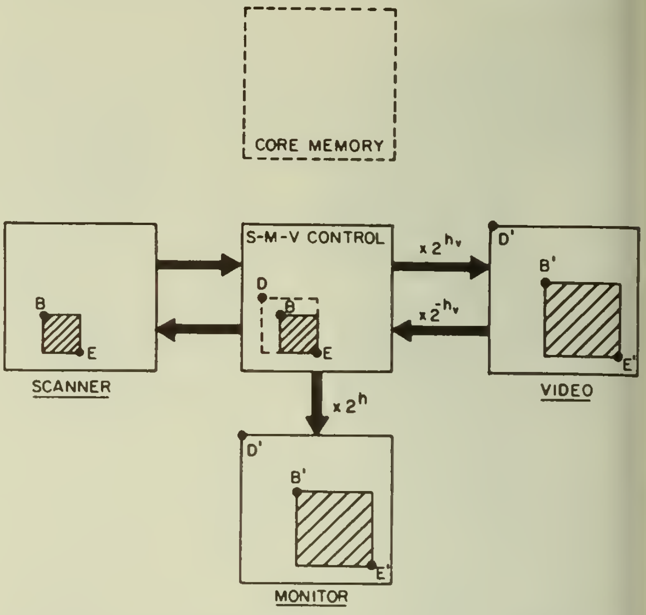


Figure 16—Transmitting between scanner and video. Display at the monitor with h = h$_v$

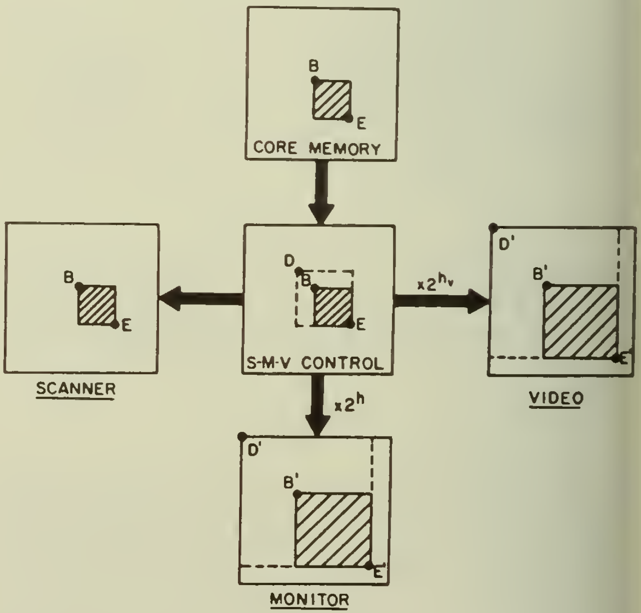


Figure 17—Transmitting from core memory to one or more media. If to video, h = h$_v$

When video is a source medium m_v is a demagnification, and when video is a destination medium m_v is a magnification.

In the (S1) interpretation of video as a source medium $\Delta X = (\Delta X)_{max}$ is assumed, and the monitor magnification is achieved in the same manner as when the source is core memory or film scanner.

Translation

When video is a destination medium translation is effected in the same way as it is at the monitor with the rotation option (R2); (XD, YD) at the S-M-V control grid transforms into (0, 0) at the video. For proper centering (XD, YD) is chosen so that:

$$m_v(|XB-XE| + 2|XD-XB|) = S(j,n) \cdot \Delta X \leq 1,$$

$$m_v(|YB-YE| + 2|YD-YB|) = \frac{\Delta Y \cdot N_u(i)}{(\Delta Y)_v} \leq 1,$$

where $N_u(i)$ is the number of useable lines per video frame (or maximum $(\Delta Y)_v$ steps).

With video as a source medium translation to the monitor is accomplished by associating (0, 0) at the video with (0, 0) at the S-M-V control grid. Translation to the monitor then takes place in the usual way. This is option (S1) and is illustrated in Figure 15. Option (S2) is accomplished by associating (0, 0) at the video with (XD, YD) at the S-M-V control grid and is illustrated in Figure 16. Translation to film then takes place as the inverse of the translation effected when the transfer is from film to video. Note that translation to monitor and film cannot be effected simultaneously; the options (S1) and (S2) are mutually exclusive as far as translation is concerned—hence the distinction.

Totally slaved as a destination

Video is not likely to be useful for display under the totally slaved concept, since this would constrain the sampling increment to be

$$\Delta X = \Delta X_{max}(j,n).$$

SUMMARY

This paper has developed the parametric description of a general purpose Scan/Display System for image digitization and display. Central to the system is the S-M-V Controller which can service either simultaneously or individually three distinct media: film, closed-circuit television and incrementally-driven CRT displays. An adjunct of the system is a Video Communications Net to provide both high and commercial resolution service to remote users.

Media compatibility

The S-M-V Controller acts as a media-media interface that identifies the necessary information transfer constraints or rejects the operation request as one demanding inconsistent parameter assignments. Transfer constraints considered include X/Y resolution ratios, aspect ratios and line sweep times of the media, and D-A/A-D conversion times. A constant data rate option allows operation at I/O channel capacity for all choices of gray scale resolution.

The digital encoding of an image generated in a scanning operation can be retransmitted to the S-M-V Controller for output (display)—and on any of the three available media.

Rasters

By associating (X,Y) positions with binary counter value pairs the controller can generate a family of the two-dimensional regular lattices—rectangular and hexagonal. X and Y resolutions are independently variable, the allowed resolution values are in geometric progression and correspond to changes in counter incrementing position. A selected "window" of the full image can be specified.

Sampling/display strategies

Three sampling formats (Raster, Coordinate and Incremental) allow a variety of sampling/display strategies. Raster format provides uniform sampling of all lattice positions. For coordinate format however, sampling takes place only at those lattice positions where the image satisfies some criteria prescribed by selection of a triggering/filtering circuit.

Incremental format is provided for segment vector plotting. Commands are provided for setting the starting point, line width and plotting resolution. Segment iteration can be specified.

The sampling beam stencil is variable in size, shape and orientation. The shape options are spot/slit. The slit option includes orientation resolution and range.

Metrological facilities

An optional local extension of position resolution can be specified through a gross/vernier counter selection. With the vernier option selected, the gross counters define a benchmark while the vernier counters represent a local displacement. Using this technique, positional resolution of 1: 30,000 is currently attainable in flying spot scanning.

Implementation

The Illiac III computer[9,10] employs a scan/display system with parameters as specified in Appendix C of this paper.

Except for the video scan converter, the microimage storage and the video storage, the scan/display system is anticipated to be operational by Summer 1969.

ACKNOWLEDGMENT

Many stimulating discussions with members of the Illiac III staff aided in the formulation of concepts developed in this paper. Mr. Robert C. Amendola has contributed significantly to the Video Network specifications and to scanner optical design. Dr. Kenneth J. Breeding participated in the first design of a scanner controller which was subsequently expanded into the S-M-V controller described in this paper.

A description of the analog and digital logic design of the scan/display system is now being prepared for publication by Dr. James L. Divilbiss and Mr. Ronald G. Martin, respectively.

The authors wish to thank Mr. John H. Otten for preparing the illustrations and Mrs. Donna J. Stutz for typing the paper.

REFERENCES

1 L A DUNN L N GOYAL B H McCORMICK V G TARESKI
S–M–V programming manual
Department of Computer Science University of Illinois Urbana Illinois March 1968

2 D M COSTIGAN
Resolution considerations affecting the electrical transmission of technical documents by scanning process
National Microfilm Association Jour Vol 1 No 3 Spring 1968

3 B F WADSWORTH
PEPR—a hardware description
Emerging Concepts in Computer Graphics Don Secrest and Jurg Nievergelt (Eds) W A Benjamin Inc New York 1968

4 R CLARK W F MILLER
Computer-based data analysis systems
Methods in Computational Physics Vol 5 Berni Adler Sidney Fernbach Manuel Rotenberg (Eds)
Academic Press New York 1966

5 R B MARR G RABINOWITZ
A software approach to the automatic scanning of digitized bubble chamber photographs
Methods in Computational Physics Vol 5 Berni Adler Sidney Fernbach Manuel Rotenberg (Eds)
Academic Press New York 1966

6 J VANDER LANS J L PELLEGRIN H J SLETTENHAAR
The hummingbird film digitizer system
SLAC Report No 82 Stanford Linear Accelerator Center Stanford University Stanford California March 1968

7 R S LEDLEY L S ROTOLO T J GOLAB J D JACOBSEN M D GINSBERG J B WILSON
FIDAC: Film input to digital automatic computer and associated syntax-directed pattern recognition programming system
Optical and Electro-optical Information Processing Teppep J Berkowitz D Clapp L Koester C and Vanderburgh Jr (Eds) M I T Press Cambridge Massachusetts 1965 Chap 33

8 H KAZMIERCZACK F HOLDERMANN
The karlsruhe system for automatic photointerpretation
Pictorial Pattern Recognition G C Cheng R S Ledley Donald K Pollock and A Rosenfeld (Eds)
Thompson Book Co Washington D C 1968

9 B H McCORMICK
Advances in the development of image processing hardware
Image Processing in Biological Science Ramsey D M (Ed) University of California Press 1968 in press

10 B H McCORMICK
The illinois pattern recognition computer—illiac III
IEEE Transactions on Electronic Computers Vol EC-12 No 5 December 1963

APPENDIX A—DEVICE SPECIFICATIONS

Scanners

The scanners are flying spot scanning systems with an added diquadrupole coil for astigmatic defocusing of the spot into a line element to achieve a slit mode. All scanners are capable of either scanning from developed film or photographing onto unexposed film. The optical path of the beam is split, with one path transversing the film and the other path through a reference grid to establish stability against engraved fiducial marks.

Several types of media transports are provided to handle the projected range of problems. A 70 mm. scanner is provided primarily for 70 mm. negative bubble chamber film. Here the format of the raster is 2.362 inches × 3.522 inches, and the minimum spot size is approximately 0.001 inch at the film. Due to the length of the frame to be scanned, scanning is done in two steps. The two horizontal halves of the frame are scanned successively with a 4 mm. overlap to establish half-frame continuity. Large motors are used for slew and gross positioning of the film and a small digital stepper motor is used for fine positioning of the frame. Frame position sensing is accomplished by using the digital stepper motor as a tachometer and by counting sprocket holes. Total film capacity is 1000 feet.

A scanner for handling 47 mm. film is similar to the 70 mm. transport design except for the following: The film format is different. A friction drive is used on the digital stepper motor, since the film is unsprocketed. The frame position is determined by sensing small index blocks at the lower edge of the film using a fibre-optics light guide and a photodiode.

The microform scanner contains three units. The first is a 35 mm. full frame digitally controlled camera which can read light through the film both negative and positive. The second unit contains a 16 mm. Bolex camera for making computer-generated black and white movies and a modified 16 mm. film editor for scanning 16 mm. film of all types. The third unit is a microfiche transport mechanism for scanning and producing a single microfiche in the 72 image COSATI format. For the three different units the C.R.T. raster is adjusted optically to fit the particular frame size.

A fourth type of scanner is built around a microscope with a digitally controlled automatic stage. Positional accuracies are on the order of ± 2 microns, and the maximum slide area coverage is 1.2 inches × 1.2 inches. Variable reduction is available from a four objective rotating turret. Full visual observation is available to an operator.

Monitors

The monitors consist of 21-inch cathode ray tubes controlled in a manner similar to the scanner C.R.T.'s; viz., digital position counters control the spot location through accurate, high-speed digital-to-analog converters. The monitor counters are digitally controlled directly from the S-M-V Controller via an incremental communications scheme; essentially the only commands issued by the S-M-V Controller for the monitors are increment the counters, decrement the counters, reset the counters, and reset the parameters. Therefore, any spot movement possible on a scanner C.R.T. can be accomplished on the monitor C.R.T. The video input for the C.R.T. grid is also synchronized by the S-M-V controller.

Included with the monitors for communications to a central processing system are a selectric typewriter, microtape input/output tape drives, and a light pen for cursor control.

Video scan converter

The video scan converter consists of a high resolution storage tube capable of storing a useable picture for at least 30 seconds. Multiple readouts can be made from a stored image before degradation is significant.

The storage tube can be written into and read from at any of the video rates in the system (525, 1536F, 1536S) on the video switching matrix side. On the SMV control side the scan converter looks like a film as seen by a scanner; therefore, reading and writing is handled in exactly the same manner as it is in a scanner.

Video switching matrix

The video switching matrix is a mechanical cross bar matrix. Therefore, the switching speed will be in the order of 100 milliseconds or less. In this routing switch any source can be switched to from one to three different destinations simultaneously. In addition, switching provisions are also included to mix any two video sources to provide a composite signal to the selected destinations.

Character generator

The Character Generator is designed to accept up to 512 ASCII characters into its 4096 bit memory. A 99 dot matrix, 9 dots wide by 11 dots high, is used to develop each character into the appropriate video levels. The maximum TV screen display is 16 horizontal rows of 32 characters or spaces each. Alternatively 132 characters/line print-out can be generated on the Videograph printer. A special cursor is also available along with eight commands for controlling it.

The output composite video signal can be either 525 or 1536 lines per frame, depending upon the externally supplied sync signal.

Videograph printer

The Videograph Printer can print on demand at a rate of 0.8 seconds per 8½ × 11 inch sheet. Horizontal resolution is 128 lines per inch and vertical resolution matches the high resolution of the 1536 line slow CCTV cameras. Gray scale resolution is limited to four shades. The paper used is inexpensive zinc-oxide coated stock.

525 line T.V. cameras and monitors

The 525 T.V. Cameras and Monitors are conventional television units; namely, 525 lines per frame, 30 frames per second interlaced (60 fields per second). These units provide for relatively low cost reduced resolution, which is sufficient for many message routing and simple acquisition and display purposes.

1536 F/S cameras

The 1536 F/S cameras are vidicon camera units which can be remotely selected to operate either in fast scan mode (15 frames per second) or slow scan mode (1.25 frames per second). The format of either mode is 1536 lines per frame done in a sequential (non-interlace) scan. The aspect ratio is variable, but it is set for a nominal 8½ × 11 aspect ratio. The camera bandwidth is limited to 9.5 Mhz for fast scan and 1.4 Mhz for slow scan.

Remote video consoles

Each remote console is a self-contained unit with two video monitors and the necessary equipment for communicating with a digital computer. The video monitors consist of a 17 inch 1536 lines per frame slow (1.25 frames per second) monitor with a P-26 phosphor and a 17 inch 1536 lines per frame fast (15 frames per second) monitor with a VC-4 phosphor. Each monitor matches characteristics of the associated camera.

Included with the console for direct digital communications to the central computer are a teletype ASR-33 unit and a small special keyboard to be used for frequently used machine orders. Other items to be included with the consoles are a microfiche reader, a digital patch panel for digital control signals, and an analog patch panel for analog control signals.

Special plug-in options for a universal cursor control could provide for such devices as a light pen, joy stick, matrix pad, bug, etc. Other options could include provisions for direct handwriting of orders at the console by T.V. camera pick-up and/or Rand tablet type device and a monitor microfiche camera for filming images from the C.R.T. screen.

Microimage storage

The Microimage storage consists of a microfiche reader/access mechanism that is able to store, retrieve, and display COSATI standard microfiche on demand. Storage of the microfiche is by a rotary drum that is a changeable unit. Images can be digitally selected, and the display of any requested image requires less than five seconds. Access time to an adjacent image (i.e., one within the same fiche) is less than two seconds. The output is displayed in an 8½ × 11 inch format if desired, and it is projected upon the two-inch vidicon of a 1536 F/S line television camera for distribution into the video network.

The drum holds 750 modified microfiche cards of 60 frames each. Each frame again contains an array of 72 microimages or basically a standard microfiche. Therefore, a single drum will provide storage for 3,240,000 page images. Readout from the selected microfiche frame is accomplished with a fly's eye readout mechanism.

Video storage

The video storage consists of an interchangeable 72 track video disk, where a single track can contain a complete image. Input and output can be at either the 1.25 or the 15 frames per second rate with resolution matched to the corresponding video devices.

APPENXIX B—SYMBOL AND PARAMETER LIST

$	Parameter values assigned at design time
*	Parameter values explicitly assigned at execution time

Upper case Latin

*	A`	Scan *A*xis selection
	ACW	*A*ngle *C*oordinate *W*ord
	B	Equivalent to (XB, YB)
*	B_s	Standard spot/nonstandard spot, or slit selection
	BCW	*B*egin *C*oordinate *W*ord
*	C	Media selection, *C*ore memory
	D	Equivalent to (XD, YD)
	DCW	*D*isplay *C*oordinate *W*ord
*	DF	*D*ata *F*ormat selection
	DPB	*D*isplay *P*arameters *B*yte
	DX	*X*-component of the incremental format *D*isplacement vector
	DY	*Y*-component of the incremental format *D*isplacement vector
	E	Equivalent to (XE, YE)
	E_v	Same as E, to distinguish *V*ernier
	ECW	*E*nd *C*oordinate *W*ord
*	F	Media selection, *F*ilm (scanner)
	FPB	*F*ormat *P*arameters *B*yte
*	G	*G*roup rotation selection for the monitor
*	J	Display regeneration request
*	K	Constant data/sample rate option (for a given p) with the raster format
*	L	*L*attice selection
	L_s	*L*ength of *S*lit = 2^v gross basic cells
	LPB	*L*attice *P*arameters *B*yte
$	N(i)	*N*umber of lines per video frame
	N_u(i)	Number of useable lines per video frame (or maximum $(\Delta Y)_v$ steps).
	N_b	*N*umber of *B*its in a raster string representation
	N_s	*N*umber of *S*amples in a raster string representation
	NGL	*N*umber of *G*ray Scale *L*evels = 2^{2^n}
	PW	*P*arameter *W*ord
*	R	Gross/vernier *R*esolution selection
*	S	*S*equence selection, sequential/interlaced

S(j,n)	Maximum achievable number of *S*amples per full video scan line
S_{max}(j)	Maximum value of S(j,n) (achieved for n = 0)
SPB	*S*lit/spot *P*arameters *B*yte
SX	*S*ign of D*X*
SY	*S*ign of D*Y*
* T	*T*rigger (filter) selection for encoding in the coordinate string representation
* V	Media selection, *V*ideo
V_s	*S*weep *V*elocity
W_s	*W*idth of *S*lit or spot diameter = 4^u gross basic cells
* XB	*X*-coordinate, *B*egin cell
* XD	*X*-coordinate, *D*isplacement
* XE	*X*-coordinate, *E*nd cell
XE_v	Same as XE, to distinguish *V*ernier
* YB	*Y*-coordinate, *B*egin cell
* YD	*Y*-coordinate, *D*isplacement
* YE	*Y*-coordinate, *E*nd cell
YE_v	Same as YE, to distinguish *V*ernier

Lower case Latin

$ b_a	Number of bits in the angle orientation counter
$ b_c	Number of interpolation bits concatenated with the b_g-p gross resolution bits
$ b_g	Number of bits in the gross position counter
$ b_a	Number of gross-vernier overlap bits
$ b_r	Number of vernier bits that extend gross resolution
$ b_v	Number of bits in the vernier position counter
c_1	Sweep velocity constant (with $c_1 = f_c$, V_s is given in basic cells per microsecond)
$ f_b	Maximum data bit transfer rate
$ f_c	Position counter incrementing frequency
f(j)	Video unit cell match function = $r_v/r_s \cdot N(i) \cdot 2^{-j}$
g(j)	Video line selection modifier. Closest interger such that f(j) $\simeq 2^{g(j)}$
* h	Monitor magnification = 2^h
$ h_{max}	Maximum value of h
$h_{v(j,n)}$	Video magnification/demagnification exponent (see m_v(j,n))
i	Video system identification
j	Video system resolution parameter, $S_{max(j)} = 2^j$
k	k(R): k = 0 for gross b_r for vernier
m	Monitor magnification = 2^h
$m_{v(j,n)}$	Video magnification/demagnification = $2^{h_v(j,n)}$
* n	2^n is the number of bits of gray scale
$ n_{max}	Maximum value of n
* p	$\Delta X = 2^p$ (see ΔX)
p_{max}(j,n)	Maximum value of p for Video Network $= b_g - j + n$
* q	$\Delta Y = 2^q$ (see ΔY)
$ r_s	Aspect ratio, control grid (*S*tandard)
$ r_v	Aspect ratio, video
* s_v	Sign bit of vernier counter
* u	Slit width is 4^u gross basic cells
$ u_{max}	Maximum value of u
* v	Slit length is 2^v gross basic cells
$ v_{max}	Maximum value of v
* z	$\Theta = 2^{z-b_a}$ (see $\Delta\Theta$)
$ z_{max}	Maximum value of z

Greek

ΔC	Counter increment along the scan axis in basic cells
$\Delta t(i)$	Time to sweep one video scan line
ΔX	Sampling increment along the X-axis at the S-M-V control grid ($\Delta X = 2^p$)
$(\Delta X)_{max}$(j,n)	Maximum value of ΔX corresponding to S(j,n)
ΔY	Sampling increment along the Y-axis at the S-M-V control grid ($\Delta Y = 2^q$)
$(\Delta Y)_v$	Sampling increment along the Y-axis at video (every $(\Delta Y)_v$ lines)
$\Delta\Theta$	Orientation sampling ($\Delta\Theta = 2^{z-b_a}$ units of angle)
δx	X-axis unit vector
δy	Y-axis unit vector
* ΘB	Orientation Begin value
* ΘE	Orientation End value

APPENDIX C—DESIGN VALUES FOR THE ILLIAC III SCAN-DISPLAY SYSTEM

b_a	angle orientation counter	8 bits
b_c	interpolation bits concatenated to gross	3 bits
b_g	gross position counter	12 bits
b_o	gross-vernier overlap	4 bits
b_r	vernier resolution extension to gross	3 bits
b_v	vernier position counter	7 bits
f_b	maximum data transfer rate	10 Mhz
f_c	counter incrementing frequency	1.25 Mhz
h_{max}	2^h is image magnification at monitor	7

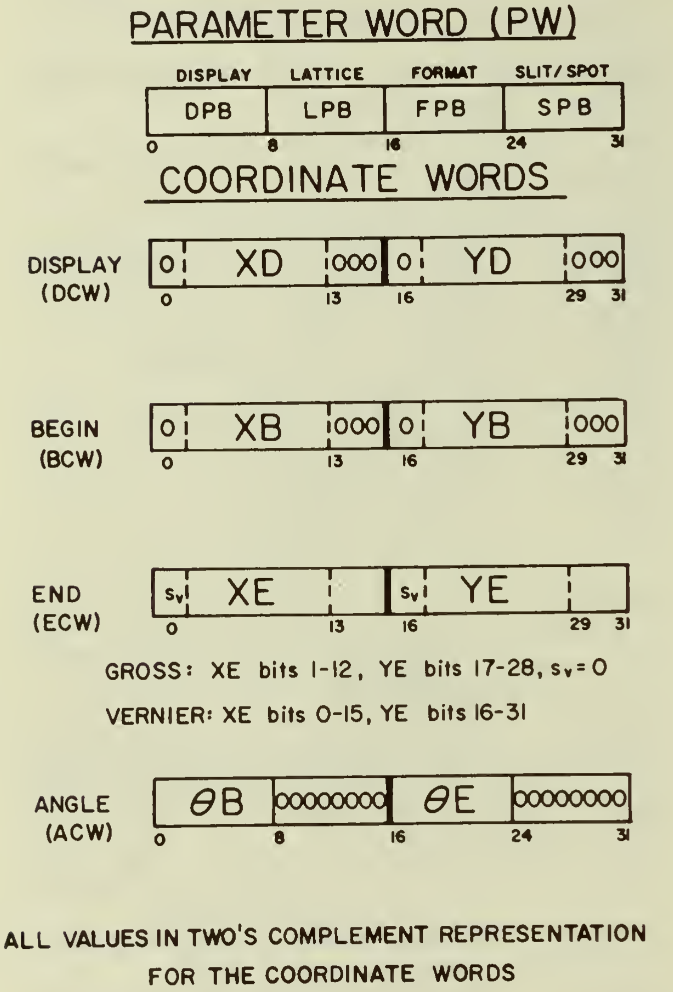
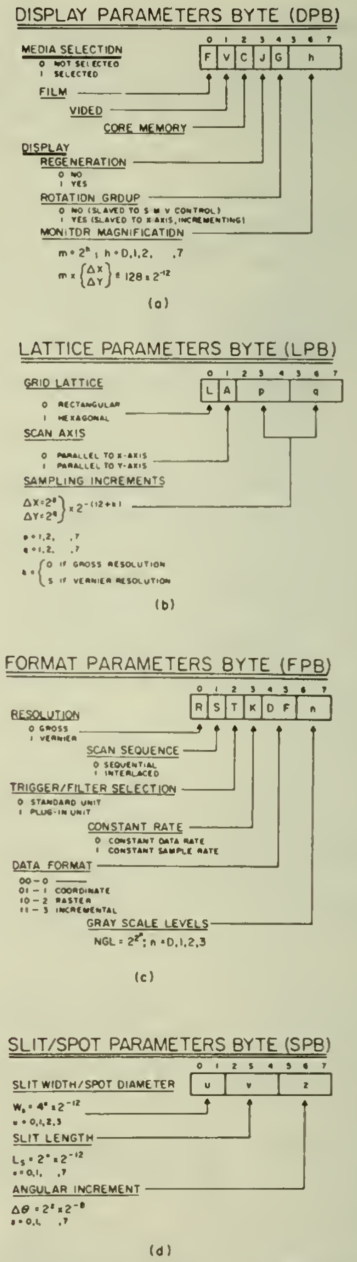


Figure 18—Parameter and coordinate words formats as defined for Illiac III

n_{max}	$2^{n_{max}}$ is maximum bits of gray scale	3
p_{max}	X-axis sample increment is 2^p basic cells	7
q_{max}	Y-axis sample increment is 2^q basic cells	7
r_s	Aspect ratio, control grid	1:1 (but variable)
r_v	Aspect ratio, Video	3:4



(a)

(b)

(c)

(d)

Figure 19—Display (a), Lattice (b), Format (c), and Slit/Spot (d) Parameter Bytes as defined for Illiac III

u_{max}	slit width is 4^u basic cells	3
v_{max}	slit length is 2^v basic cells	7
z_{max}	angle increment is 2^{z-b_a} units	7

Form AEC-427
(6/68)
AECM 3201

U.S. ATOMIC ENERGY COMMISSION
UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO. 473 COO-2118-0021	2. TITLE ILLIAC III REFERENCE MANUAL VOLUME III: Input/Output

3. TYPE OF DOCUMENT (Check one):

☒ a. Scientific and technical report
□ b. Conference paper not to be published in a journal:
Title of conference ________
Date of conference ________
Exact location of conference ________
Sponsoring organization ________
□ c. Other (Specify) ________

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a. AEC's normal announcement and distribution procedures may be followed.
□ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
□ c. Make no announcement or distrubution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)
Professor B. H. McCormick
Principal Investigator
Illiac III Project

Organization
Department of Computer Science
University of Illinois
Urbana, Illinois

Signature Bruce H. McCormick	Date August 13, 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

□ a. AEC patent clearance has been granted by responsible AEC patent group.
□ b. Report has been sent to responsible AEC patent group for clearance.
□ c. Patent clearance not required.